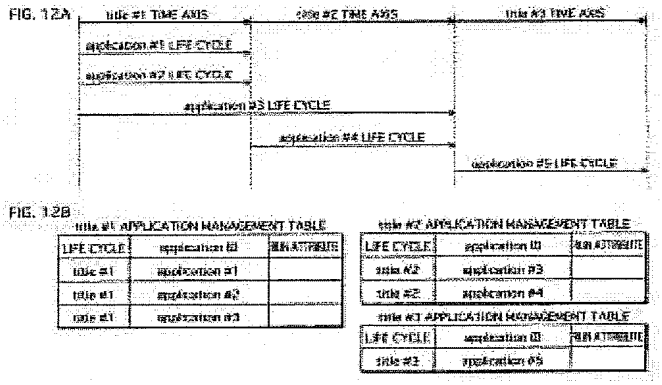
RECORDING MEDIUM, REPRODUCTION DEVICE, PROGRAM, AND REPRODUCTION METHOD

Publication number: JP4091078 (B2)
Publication date: 2008-05-28
Inventor(s):
Applicant(s):
Classification:
- international: ***G11B20/10; G06F9/06; G06F19/00; G11B20/12; G11B27/00; G11B27/10; G11B27/32; H04N5/93;*** **G11B20/10; G06F9/06; G06F19/00; G11B20/12; G11B27/00; G11B27/10; G11B27/32; H04N5/93**
- European: G11B20/10; G11B27/10A1; G11B27/32D2
Application number: JP20050514678T 20041012
Priority number(s): JP20030352913 20031010; JP20030379758 20031110; WO2004JP15333 20041012

Also published as:
EP1672637 (A1)
US2006282612 (A1)
US2007089156 (A1)
US2007274680 (A1)
US2007089146 (A1)
more >>

Abstract not available for JP 4091078 (B2)
Abstract of corresponding document: **EP 1672637 (A1)**

A BD-ROM contains a plurality of titles which can be branched among, and a Java application. The Java application is a program described in a programming language for a virtual machine. A life cycle where execution by the virtual machine is enabled is predetermined. Each of the titles contains an application management table indicates an application that has a life cycle bound to the title.



(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4091078号
(P4091078)
(45)発行日　平成20年5月28日(2008.5.28)　(24)登録日　平成20年3月7日(2008.3.7)

(51)Int.Cl.		F I		
G11B 20/10	(2006.01)	G11B	20/10	321Z
G11B 27/00	(2006.01)	G11B	27/00	D
H04N 5/93	(2006.01)	H04N	5/93	Z

請求項の数 4　(全 61 頁)

(21)出願番号　特願2005-514678 (P2005-514678)
(86)(22)出願日　平成16年10月12日 (2004.10.12)
(86)国際出願番号　PCT/JP2004/015333
(87)国際公開番号　WO2005/036545
(87)国際公開日　平成17年4月21日 (2005.4.21)
審査請求日　平成19年10月11日 (2007.10.11)
(31)優先権主張番号　特願2003-352913 (P2003-352913)
(32)優先日　平成15年10月10日 (2003.10.10)
(33)優先権主張国　日本国 (JP)
(31)優先権主張番号　特願2003-379758 (P2003-379758)
(32)優先日　平成15年11月10日 (2003.11.10)
(33)優先権主張国　日本国 (JP)

早期審査対象出願

(73)特許権者　000005821
松下電器産業株式会社
大阪府門真市大字門真１００６番地
(74)代理人　100090446
弁理士　中島　司朗
(72)発明者　池田　航
大阪府門真市大字門真１００６番地　松下電器産業株式会社内
(72)発明者　岩本　啓明
大阪府門真市大字門真１００６番地　松下電器産業株式会社内
(72)発明者　岡田　智之
大阪府門真市大字門真１００６番地　松下電器産業株式会社内

最終頁に続く

(54)【発明の名称】記録媒体、再生装置、記録方法、再生方法

(57)【特許請求の範囲】

【請求項１】

インデックステーブルと、動作モードオブジェクトと、AVストリームと、プレイリスト情報とが記録された記録媒体であって、

前記インデックステーブルは、タイトルと、動作モードオブジェクトとの対応付けを示すとともに、各タイトルに対応する動作モードオブジェクトが、ムービーモード用及び仮想マシンモード用の何れであるかを示し、

ムービモード用の動作モードオブジェクトは、制御手順を表すナビゲーションコマンドを含み、

プレイリスト情報は、経路情報を含み、経路情報は、再生の対象となるAVストリームの指定と、そのAVストリームの再生時間軸における開始時間情報及び終了時間情報の組みとを含み、

前記アプリケーションは、プレイリスト情報についてのプレーヤインスタンスを生成することで、前記AVストリームの再生を再生装置に行わせるプログラムであり、

仮想マシンモード用の動作モードオブジェクトは、キャッシュ管理情報を含み、

前記キャッシュ管理情報は、

仮想マシンモード用の動作モードオブジェクトに対応するタイトルが、これから再生しようとするタイトルであるカレントタイトルになった際、当該タイトルにおけるAV再生が開始されるまでに、どのアプリケーションを構成するファイルを、キャッシュに読み込むべきかを示す

ことを特徴とする記録媒体。

【請求項２】

記録媒体に記録されたタイトルを再生すると共に、アプリケーションを実行する再生装置であって、

インデックステーブルに基づき、複数タイトルの中から、タイトルを選ぶモジュールマネージャと、

アプリケーションを実行するモジュールとを備え、

前記記録媒体には、インデックステーブルと、動作モードオブジェクトと、AVストリームと、プレイリスト情報とが記録され、

前記インデックステーブルは、タイトルと、動作モードオブジェクトとの対応付けを示すとともに、各タイトルに対応する動作モードオブジェクトが、ムービーモード用及び仮想マシンモード用の何れであるかを示し、

ムービモード用の動作モードオブジェクトは、制御手順を表すナビゲーションコマンドを含み、

プレイリスト情報は、経路情報を含み、経路情報は、再生の対象となるAVストリームの指定と、そのAVストリームの再生時間軸における開始時間情報及び終了時間情報の組みとを含み、

前記アプリケーションは、プレイリスト情報についてのプレーヤインスタンスを生成することで、前記AVストリームの再生を再生装置に行わせるプログラムであり、

仮想マシンモード用の動作モードオブジェクトは、キャッシュ管理情報を含み、

前記キャッシュ管理情報は、

仮想マシンモード用の動作モードオブジェクトに対応するタイトルが、これから再生しようとするタイトルであるカレントタイトルになった際、当該タイトルにおけるAV再生が開始されるまでに、どのアプリケーションを構成するファイルを、キャッシュに読み込むべきかを示す

前記モジュールは、

カレントタイトルの選択があった場合、当該タイトルにおけるAV再生が開始されるまでに、カレントタイトルに対応する動作モードオブジェクト内のキャッシュ管理情報に示されているファイルを、キャッシュに読み込む

ことを特徴とする再生装置。

【請求項３】

ボリュームデータを生成して、ボリュームデータが記録された記録媒体を得る記録方法であって、

ボリュームデータは、インデックステーブルと、動作モードオブジェクトと、AVストリームと、プレイリスト情報とを含み、

前記インデックステーブルは、タイトルと、動作モードオブジェクトとの対応付けを示すとともに、各タイトルに対応する動作モードオブジェクトが、ムービーモード用及び仮想マシンモード用の何れであるかを示し、

ムービモード用の動作モードオブジェクトは、制御手順を表すナビゲーションコマンドを含み、

プレイリスト情報は、経路情報を含み、経路情報は、再生の対象となるAVストリームの指定と、そのAVストリームの再生時間軸における開始時間情報及び終了時間情報の組みとを含み、

前記アプリケーションは、プレイリスト情報についてのプレーヤインスタンスを生成することで、前記AVストリームの再生を再生装置に行わせるプログラムであり、

仮想マシンモード用の動作モードオブジェクトは、キャッシュ管理情報を含み、

前記キャッシュ管理情報は、

仮想マシンモード用の動作モードオブジェクトに対応するタイトルが、これから再生しようとするタイトルであるカレントタイトルになった際、当該タイトルにおけるAV再生が開始されるまでに、どのアプリケーションを構成するファイルを、キャッシュに読み込む

べきかを示す
ことを特徴とする記録方法。
【請求項４】
記録媒体に記録されたタイトルを再生すると共に、アプリケーションを実行する再生方法であって、
インデックステーブルに基づき、複数タイトルの中から、タイトルを選ぶマネージャステップと、
アプリケーションを実行するモジュールステップとを含み、
前記記録媒体には、インデックステーブルと、動作モードオブジェクトと、AVストリームと、プレイリスト情報とが記録され、
前記インデックステーブルは、タイトルと、動作モードオブジェクトとの対応付けを示すとともに、各タイトルに対応する動作モードオブジェクトが、ムービーモード用及び仮想マシンモード用の何れであるかを示し、
ムービモード用の動作モードオブジェクトは、制御手順を表すナビゲーションコマンドを含み、
プレイリスト情報は、経路情報を含み、経路情報は、再生の対象となるAVストリームの指定と、そのAVストリームの再生時間軸における開始時間情報及び終了時間情報の組みとを含み、
前記アプリケーションは、プレイリスト情報についてのプレーヤインスタンスを生成することで、前記AVストリームの再生を再生装置に行わせるプログラムであり、
仮想マシンモード用の動作モードオブジェクトは、キャッシュ管理情報を含み、
前記キャッシュ管理情報は、
仮想マシンモード用の動作モードオブジェクトに対応するタイトルが、これから再生しようとするタイトルであるカレントタイトルになった際、当該タイトルにおけるAV再生が開始されるまでに、どのアプリケーションを構成するファイルを、キャッシュに読み込むべきかを示し
前記モジュールステップは、
カレントタイトルの選択があった場合、当該タイトルにおけるAV再生が開始されるまでに、カレントタイトルに対応する動作モードオブジェクト内のキャッシュ管理情報に示されているファイルを、キャッシュに読み込む
ことを特徴とする再生方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、デジタル化された映画作品の再生と、アプリケーションの実行とを同時に実行する、再生制御技術の技術分野に属する発明であり、本再生制御技術を民生用の再生装置、プログラムに応用する場合の応用技術に深く係る。
【背景技術】
【０００２】
上述したような同時実行させるにあたっては、デジタルストリームの時間軸においてアプリケーションの生存区間をどう定めるかが問題になる。アプリケーションの生存区間とは、アプリケーションによるサービスを開始してから、サービスを終了するまでの期間をいう。かかる期間をチャプターのように細かい単位に定めれば、様々なアプリケーションによるサービスをかわるがわる実行することができる。
【０００３】
かかる背景技術には、以下に示される文献公知発明が存在する。
【特許文献１】
特許第2813245号
【発明の開示】
【発明が解決しようとする課題】

【０００４】
しかしアプリケーションの生存区間を細かく定めれば、記録媒体からのアプリケーション読み出し回数も増える。一方、BD-ROM、DVDのような映画作品頒布用の光ディスク媒体は、概して読出速度が遅い。このような遅い速度による読み出しの回数が増えれば、映画作品本編を構成するビデオストリームの読み出しに影響が生じ、動画再生が途切れがちになる。多様なサービスが実現されるとはいえ、動画再生の妨げがあるような同時実行では、ユーザを大きく落胆させることになり、また映画作品の制作者からは、敬遠されることになる。

【０００５】
本発明の目的は、アプリケーションの生存区間を細かい単位に定めることができる記録媒体を提供することである。

【課題を解決するための手段】

【０００６】
上記課題を解決するため、本発明にかかる記録媒体は、インデックステーブルと、動作モードオブジェクトとが記録された記録媒体であって、前記インデックステーブルは、タイトルと、動作モードオブジェクトとの対応付けを示すとともに、各タイトルに対応する動作モードオブジェクトが、ムービーモード用及び仮想マシンモード用の何れであるかを示し、前記動作モードオブジェクトには、ムービモード用のものと、仮想マシンモード用のものとがあり、ムービモード用の動作モードオブジェクトは、制御手順を表すナビゲーションコマンドを含み、仮想マシンモード用の動作モードオブジェクトは、キャッシュ管理情報を含み、前記キャッシュ管理情報は、仮想マシンモード用の動作モードオブジェクトに対応するタイトルが、これから再生しようとするタイトルであるカレントタイトルになった際、当該タイトルにおけるAV再生が開始されるまでに、どのアプリケーションを構成するファイルを、キャッシュに読み込むべきかを示すことを特徴としている。

【０００７】
本発明にかかる記録媒体において、キャッシュ管理情報は、動作モードオブジェクトに対応するタイトルがカレントタイトルになった際、当該タイトルにおけるAV再生が開始されるまでに、どのアプリケーションを構成するファイルを、キャッシュに読み込むべきかを示すので、動作モードオブジェクトに対応するタイトルがカレントタイトルになった場合にあっては、アプリケーションがいつでも、キャッシュから仮想マシンにアプリケーションを読み出せる状態になる。仮想マシンへのアプリケーション読み出しはいつでも行なえるので、チャプターといった細かい単位でアプリケーションの生存区間を定めたとしても、記録媒体からのアプリケーションの読み出し回数を減らすことができる。また、光ディスクからキャッシュへの読み出しは、シームレス再生の保証が不要な、タイトル分岐時になされるので、アプリケーション読み出しによる中断をユーザに意識させることなく、アプリケーションをいつでも実行できる容易に準備しておくことができる。

【００１２】
（第１実施形態）
以降、本発明に係る再生装置の実施形態について説明する。先ず始めに、本発明に係る再生装置の実施行為のうち、使用行為についての形態を説明する。図１は、本発明に係る再生装置の、使用行為についての形態を示す図である。図１において、本発明に係る再生装置は再生装置２００であり、テレビ３００、リモコン４００と共にホームシアターシステムを形成する。

【００１３】
このBD-ROM１００は、再生装置２００、リモコン３００、テレビ４００により形成されるホームシアターシステムに、映画作品を供給するという用途に供される。
以上が本発明に係る再生装置の使用形態についての説明である。
続いて本発明に係る再生装置の再生の対象となる、記録媒体であるBD-ROMについて説明する。BD-ROMにより、ホームシアターシステムに供給されるディスクコンテンツは、互いに分岐可能な複数タイトルから構成される。各タイトルは、1つ以上のプレイリストと、

このプレイリストを用いた動的な制御手順とからなる。

【００１４】

プレイリストとは、1つ以上のデジタルストリームと、そのデジタルストリームにおける再生経路とから構成され、”時間軸”の概念をもつBD-ROM上のアクセス単位である。以上のプレイリストと、動的な制御手順とを包含しているため、タイトルはデジタルストリーム特有の時間軸の概念と、コンピュータプログラム的な性質とを併せもっている。

図２は、BD-ROMにおけるファイル・ディレクトリ構成を示す図である。本図においてBD-ROMには、Rootディレクトリの下に、BDMVディレクトリがある。

【００１５】

BDMVディレクトリには、拡張子bdmvが付与されたファイル(index.bdmv,MovieObject.bdmv)と、拡張子BD-Jが付与されたファイル(00001.BD-J,00002.BD-J,00003.BD-J)がある。そしてこのBDMVディレクトリの配下には、更にPLAYLISTディレクトリ、CLIPINFディレクトリ、STREAMディレクトリ、BDARディレクトリと呼ばれる4つのサブディレクトリが存在する。

【００１６】

PLAYLISTディレクトリには、拡張子mplsが付与されたファイル(00001.mpls,00002.mpls,00003mpls)がある。

CLIPINFディレクトリには、拡張子clpiが付与されたファイル(00001.clpi,00002.clpi,00003.clpi)がある。

STREAMディレクトリには、拡張子m2tsが付与されたファイル(00001.m2ts,00002.m2ts,00003.m2ts)がある。

【００１７】

BDARディレクトリには、拡張子jarが付与されたファイル(00001.jar,00002.jar,00003jar)がある。以上のディレクトリ構造により、互いに異なる種別の複数ファイルが、BD-ROM上に配置されていることがわかる。

本図において拡張子.m2tsが付与されたファイル(00001.m2ts,00002.m2ts,00003.m2ts・・・・・)は、AVClipを格納している。AVClipには、MainCLip、SubClipといった種別がある。MainCLipは、ビデオストリーム、オーディオストリーム、プレゼンテーショングラフィクスストリーム、インタラクティブグラフィクスストリームというような複数エレメンタリストリームを多重化することで得られたデジタルストリームである。

【００１８】

SubClipは、オーディオストリーム、グラフィクスストリーム、テキスト字幕ストリーム等、1つのエレメンタリストリームのみにあたるデジタルストリームである。

拡張子”clpi”が付与されたファイル(00001.clpi,00002.clpi,00003.clpi・・・・・)は、AVClipのそれぞれに1対1に対応する管理情報である。管理情報故に、Clip情報は、AVClipにおけるストリームの符号化形式、フレームレート、ビットレート、解像度等の情報や、頭出し位置を示すEP_mapをもっている。

【００１９】

拡張子”mpls”が付与されたファイル(00001.mpls,00002.mpls,00003.mpls・・・・・)は、プレイリスト情報を格納したファイルである。プレイリスト情報は、AVClipを参照してプレイリストを定義する情報である。プレイリストは、MainPath情報、PLMark情報、SubPath情報から構成される。

MainPath情報は、複数のPlayItem情報からなる。PlayItemとは、1つ以上のAVClip時間軸上において、In_Time,Out_Timeを指定することで定義される再生区間である。PlayItem情報を複数配置させることで、複数再生区間からなるプレイリスト(PL)が定義される。図３は、AVClipと、PLとの関係を示す図である。第1段目はAVClipがもつ時間軸を示し、第2段目は、PLがもつ時間軸を示す。PL情報は、PlayItem#1,#2,#3という3つのPlayItem情報を含んでおり、これらPlayItem#1,#2,#3のIn_time,Out_timeにより、3つの再生区間が定義されることになる。これらの再生区間を配列させると、AVClip時間軸とは異なる時間軸が定義されることになる。これが第2段目に示すPL時間軸である。このように、PlayItem

情報の定義により、AVClipとは異なる時間軸の定義が可能になる。

【００２０】

AVClipに対する指定は、原則1つであるが、複数AVClipに対する一括指定もあり得る。この一括指定は、PlayItem情報における複数のClip_Information_file_nameによりなされる。図４は、4つのClip_Information_file_nameによりなされた一括指定を示す図である。本図において第1段目～第4段目は、4つのAVClip時間軸(AVClip#1,#2,#3,#4の時間軸)を示し、第5段目は、PL時間軸を示す。PlayItem情報が有する、4つのClip_Information_file_nameにて、これら4つの時間軸が指定されている。こうすることで、PlayItemが有するIn_time,Out_timeにより、択一的に再生可能な4つの再生区間が定義されることになる。これにより、PL時間軸には、切り換え可能な複数アングル映像からなる区間(いわゆるマルチアングル区間)が定義されることになる。

【００２１】

PLmark情報は、PL時間軸のうち、任意の区間を、チャプターとして指定する情報である。図５は、PLmarkによるチャプター定義を示す図である。本図において第1段目は、AVClip時間軸を示し、第2段目はPL時間軸を示す。図中の矢印pk1,2は、PLmarkにおけるPlayItem指定(ref_to_PlayItem_Id)と、一時点の指定(mark_time_stamp)とを示す。これらの指定によりPL時間軸には、3つのチャプター(Chapter#1,#2,#3)が定義されることになる。

【００２２】

SubPath情報は、複数のSubPlayItem情報からなる。SubPlayItem情報は、SubClipの時間軸上にIn_Time,Out_Timeを指定することで再生区間を定義する。またSubPlayItem情報は、SubClip時間軸上の再生区間を、PL時間軸に同期させるという同期指定が可能であり、この同期指定により、PL時間軸と、SubPlayItem情報時間軸とは同期して進行することになる。図６は、SubPlayItem時間軸上の再生区間定義と、同期指定を示す図である。本図において第1段目は、PL時間軸を示し、第2段目はSubPlayItem時間軸を示す。図中のSubPlayItem.IN_timeは再生区間の始点を、SubPlayItem.Out_timeは再生区間の終点をそれぞれ示す。これによりSubClip時間軸上にも再生区間が定義されていることがわかる。矢印Sn1においてSync_PlayItem_Idは、PlayItemに対する同期指定を示し、矢印Sn2においてsync_start_PTS_of_PlayItemは、PL時間軸におけるPlayItem上の一時点の指定を示す。

【００２３】

複数AVClipの切り換えを可能とするマルチアングル区間や、AVClip－SubClipを同期させ得る同期区間の定義を可能とするのが、BD-ROMにおけるプレイリスト情報の特徴である。以上のClip情報及びプレイリスト情報は、”静的シナリオ”に分類される。何故なら、以上のClip情報及びプレイリスト情報により、静的な再生単位であるPLが定義されるからである。以上で静的シナリオについての説明を終わる。

【００２４】

続いて”動的なシナリオ”について説明する。動的シナリオとは、AVClipの再生制御を動的に規定するシナリオデータである。”動的に”というのは、再生装置における状態変化やユーザからのキーイベントにより再生制御の中身がかわることをいう。BD-ROMでは、この再生制御の動作環境として2つのモードを想定している。1つ目は、DVD再生装置の動作環境と良く似た動作環境であり、コマンドベースの実行環境である。2つ目は、Java仮想マシンの動作環境である。これら2つの動作環境のうち1つ目は、HDMVモードと呼ばれる。2つ目は、BD-Jモードと呼ばれる。これら2つの動作環境があるため、動的シナリオはこのどちらかの動作環境を想定して記述される。HDMVモードを想定した動的シナリオはMovieオブジェクトと呼ばれ、管理情報により定義される。一方BD-Jモードを想定した動的シナリオはBD-Jオブジェクトと呼ばれる。

【００２５】

先ず初めにMovieオブジェクトについて説明する。

＜Movieオブジェクト＞

Movieオブジェクトは、”タイトル”の構成要素であり、ファイルMovieObject.bdmvに格納される。図７（ａ）は、Movieオブジェクトの内部構成を示す図である。Movieオブジ

エクトは、属性情報、複数のナビゲーションコマンドからなるコマンド列からなる。

【００２６】

属性情報は、PL時間軸において、MenuCallがなされた際、MenuCall後の再生再開を意図しているか否かを示す情報(resume_intention_flag)、PL時間軸においてMenuCallをマスクするかを示す情報(menu_call_mask)、タイトルサーチをマスクするかを示す情報(title_search_flag)からなる。Movieオブジェクトは、”時間軸”+”プログラム的制御”という2つの性質を併せ持つことができるで、本編再生を実行するもの等、多様な種類のタイトルがこのMovieオブジェクトにより記述されることになる。

【００２７】

ナビゲーションコマンド列は、条件分岐、再生装置における状態レジスタの設定、状態レジスタの設定値取得等を実現するコマンド列からなる。Movieオブジェクトにおいて記述可能なコマンドを以下に示す。

PlayPLコマンド

書式:PlayPL(第１引数，第２引数)

第１引数は、プレイリストの番号で、再生すべきPLを指定することができる。第２引数は、そのPLに含まれるPlayItemや、そのPLにおける任意の時刻、Chapter、Markを用いて再生開始位置を指定することができる。

【００２８】

PlayItemによりPL時間軸上の再生開始位置を指定したPlayPL関数をPlayPLatPlayItem()、

Chapterにより PL時間軸上の再生開始位置を指定したPlayPL関数をPlayPLatChapter()、

時刻情報により PL時間軸上の再生開始位置を指定したPlayPL関数をPlayPLatSpecified Time()という。

JMPコマンド

書式：JMP 引数

JMPコマンドは、現在の動的シナリオを途中で廃棄し(discard)、引数たる分岐先動的シナリオを実行するという分岐である。JMP命令の形式には、分岐先動的シナリオを直接指定している直接参照のものと、分岐先動的シナリオを間接参照している間接参照のものがある。

Movieオブジェクトにおけるナビゲーションコマンドの記述は、DVDにおけるナビゲーションコマンドの記述方式と良く似ているので、DVD上のディスクコンテンツを、BD-ROMに移植するという作業を効率的に行うことができる。Movieオブジェクトについては、以下の国際公開公報に記載された先行技術が存在する。詳細については、本国際公開公報を参照されたい。

国際公開公報WO 2004/074976

以上でMovieオブジェクトについての説明を終える。続いてBD-Jオブジェクトについて説明する。

【００２９】

＜BD-Jオブジェクト＞

拡張子BD-Jが付与されたファイル(00001.BD-J,00002.BD-J,00003.BD-J)は、BD-Jオブジェクトを構成する。BD-Jオブジェクトは、Javaプログラミング環境で記述された、BD-Jモードの動的シナリオである。図７（ｂ）は、BD-Jオブジェクトの内部構成を示す図である。本図に示すようにBD-Jオブジェクトは、Movieオブジェクト同様の属性情報、アプリケーション管理テーブルからなる。属性情報を有している点でBD-JオブジェクトはMovieオブジェクトとほぼ同じである。Movieオブジェクトとの違いは、BD-Jオブジェクトはコマンドが直接記述されていない点である。つまりMovieオブジェクトにおいて制御手順は、ナビゲーションコマンドにより直接記述されていた。これに対しBD-Jオブジェクトでは、そのタイトルを生存区間としているJavaアプリケーションをアプリケーション管理テーブル上に定めることにより、間接的に制御手順を規定している。このような間接的な規定により、複数タイトルにおいて制御手順を共通化するという、制御手順の共通化を効率的に

行うことができる。

【００３０】

図７（ｃ）は、Javaアプリケーションの内部構成を示す図である。本図においてアプリケーションは、仮想マシンのヒープ領域(ワークメモリとも呼ばれる)にロードされた1つ以上のxletプログラムからなる。このワークメモリでは、1つ以上のスレッドが動作しており、ワークメモリにロードされたxletプログラム、及び、スレッドから、アプリケーションは構成されることになる。以上がアプリケーションの構成である。

【００３１】

このアプリケーションの実体にあたるのが、BDMVディレクトリ配下のBDARディレクトリに格納されたJavaアーカイブファイル(00001.jar,00002.jar)である。以降、Javaアーカイブファイルについて説明する。

Javaアーカイブファイル(00001.jar,00002.jar)は、Javaアプリケーションを構成するプログラム、データを格納したアーカイブファイルである。図８（ａ）は、アーカイブファイルにより収められているプログラム、データを示す図である。本図におけるデータは、枠内に示すディレクトリ構造が配置された複数ファイルを、javaアーカイバでまとめたものである。枠内に示すディレクトリ構造は、rootディレクトリ、javaディレクトリ、imageディレクトリとからなり、rootディにcommon.pkgが、javaディレクトリにaaa.class,bbb.classが、imageディレクトリに、menu.jpgが配置されている。javaアーカイブファイルは、これらをjavaアーカイバでまとめることで得られる。かかるデータは、BD-ROMからキャッシュに読み出されるにあたって展開され、キャッシュ上で、ディレクトリに配置された複数ファイルとして取り扱われる。Javaアーカイブファイルのファイル名における"xxxxx"という5桁の数値は、アプリケーションのID(applicationID)を示す。本Javaアーカイブファイルがキャッシュに読み出された際、このファイル名における数値を参照することにより、任意のJavaアプリケーションを構成するプログラム，データを取り出すことができる。

【００３２】

Javaアーカイブファイルにおいて1つにまとめられるファイルには、xletプログラムがある。

xletプログラムは、JMF(Java Media FrameWork)インターフェイスを利用することができるJavaプログラムである。xletプログラムは、キーイベントを受信するEventListner等、複数の関数からなり、JMF等の方式に従って、受信したキーイベントに基づく処理を行う。

【００３３】

図８（ｂ）は、xletプログラムの一例を示す図である。JMF A" BD://00001.mpls" ;は、PLを再生するプレーヤインスタンスの生成をJava仮想マシンに命じるメソッドである。A.playは、JMFプレーヤインスタンスに再生を命じるメソッドである。かかるJMFプレーヤインスタンス生成は、JMFライブラリに基づきなされる。xletプログラムの記述は、BD-ROMのPLに限らず、時間軸をもったコンテンツ全般に適用可能なJMFの記述である。このような記述が可能であるので、Javaプログラミングに長けたソフトハウスに、BD-Jオブジェクト作成を促すことができる。

【００３４】

図８（ｂ）におけるJumpTItle();は、ファンクションAPIのコールである。このファンクションAPIは、他のタイトルへの分岐(図中ではtitle#1)を再生装置に命じるものである。ここでファンクションAPIとは、BD-ROM再生装置により供給されるAPI(Appliation Interface)である。JumpTitleコマンドの他にも、ファンクションAPIのコールにより、BD-ROM再生装置特有の処理をxletプログラムに記述することができる。

【００３５】

BD-JモードにおいてPL再生は、JMFインターフェイスにより規定される。このJMFプレーヤインスタンスは、PL時間軸を定めるものだから、タイトル時間軸は、このJMFプレーヤインスタンスをもったタイトルから定まる。また

BD-Jモードにおいてタイトルからタイトルへの分岐はJumpTitleAPIのコールにより規定される。JumpTitleAPIコールは、いわばタイトルの終了時点を定めるものなので、こうしたJMFプレーヤインスタンス、JumpTitleAPIコールをもったアプリケーションが、BD-Jモードにおいてタイトルの開始及び終了を律することになる。かかるアプリケーションを本編再生アプリケーションという。

【００３６】

以上が、BD-Jモードにおける動的シナリオについての説明である。このBD-Jモードにおける動的シナリオにより、PL再生と、プログラム的制御とを併せもったタイトルが定義されることになる。尚、本実施形態においてアプリケーションを構成するプログラム、データは、Javaアーカイブファイルにまとめられたが、LZHファイル、zipファイルであってもよい。

【００３７】

＜タイトル時間軸＞

タイトルを構成する静的シナリオ、動的シナリオについて説明を終えたところで、これらによりどのような時間軸が定義されるかについて説明する。タイトルにより定義される時間軸は、”タイトル時間軸”と呼ばれる。タイトル時間軸とは、Movieオブジェクト、又は、BD-Jオブジェクトにより再生が命じられるPLにより構成される。ここで一例を挙げるのは、図９（ａ）のようなタイトルである。このタイトルは、トップメニュー→title#1→title#2→トップメニュー、トップメニュー→title#3→トップメニューという一連のタイトルである。かかるタイトルのうち、title#1はPlayList#1、PlayList#2、title#2がPlayList#3、title#3がPlayList#4の再生を命じるものなら、図９（ｂ）のように、PlayList#1、PlayList#2の時間軸を足し合わせた時間軸を、title#1はもつことになる。同様にtitle#2は、PlayList#3時間軸からなる時間軸を、title#3はPlayList#4時間軸からなる時間軸を持つことになる。これらタイトル時間軸におけるPL時間軸ではシームレス再生が保証されるが、タイトル時間軸間ではシームレス再生の保証は必要でなくなる。Javaアプリケーションを動作させるにあたっては、Javaアプリケーションを、仮想マシンのワークメモリ上に存在させてもよい期間(サービス期間)を、こうしたタイトル時間軸上に定義せねばならない。BD-Jモードにおいて Javaアプリケーションを動作させるにあたっては、互いに分岐し合う時間軸上に、Javaアプリケーションのサービス期間を定義せねばならない。このサービス期間の定義が、BD-ROM向けのプログラミングを行うにあたっての留意点になる。

【００３８】

最後に、index.bdmvに格納されたIndexTableについて説明する。IndexTableは、タイトル番号と、Movieオブジェクト、BD-Jオブジェクトとを対応づけるテーブルであり、動的シナリオから動的シナリオへの分岐の際、参照される間接参照用テーブルである。IndexTableは、複数ラベルのそれぞれに対するIndexからなる。各Indexには、そのラベルに対応する動的シナリオの識別子が記述されている。こうしたIndexTableを参照することで、Movieオブジェクト、BD-Jオブジェクトの違いを厳密に区別することなく、分岐を実現することができる。IndexTableについては、以下の国際公開公報に詳細が記載されている。詳細については、本公報を参照されたい。

国際公開公報WO 2004/025651 A1公報

以上がBD-ROMに記録されているファイルについて説明である。

【００３９】

＜アプリケーション管理テーブル＞

JMFプレーヤインスタンス、JumpTitleAPIコールをもったアプリケーションが、タイトル時間軸を律することは上述した通りだが、JMFプレーヤインスタンスやJumpTitleAPIのコールをもたないその他のアプリケーションを、タイトル時間軸上で動作させる場合、時間軸の何処からアプリケーションによるサービスを開始し、時間軸の何処でアプリケーションによるサービスを終えるかという”サービスの開始点・終了点”を明確に規定することが重要になる。本実施形態では、アプリケーションによるサービスが開始してから、終

了するまでを、”アプリケーションの生存”として定義する。アプリケーションの生存を定義するための情報は、BD-Jオブジェクトにおけるアプリケーション管理テーブルに存在する。以降アプリケーション管理テーブルについてより詳しく説明する。

【００４０】

アプリケーション管理テーブル(AMT)は、各タイトルが有しているタイトル時間軸において、仮想マシンのワークメモリ上で生存し得るアプリケーションを示す情報である。ワークメモリにおける生存とは、そのアプリケーションを構成するxletプログラムが、ワークメモリに読み出され、仮想マシンによる実行が可能になっている状態をいう。図７（ｂ）における破線矢印at1は、アプリケーション管理テーブルの内部構成をクローズアップして示す。この内部構成に示すように、アプリケーション管理テーブルは、『生存区間』と、そのタイトルを生存区間としているアプリケーションを示す『applicationID』と、そのアプリケーションの『起動属性』とからなる。

【００４１】

近い将来、実施されるであろうディスクコンテンツを題材に選んで、アプリケーション管理テーブルにおける生存区間記述について、具体例を交えて説明する。ここで題材にするディスクコンテンツは、映像本編を構成する本編タイトル(title#1)、オンラインショッピングを構成するオンラインショッピングタイトル(title#2)、ゲームアプリケーションを構成するゲームタイトル(title#3)という、性格が異なる3つのタイトルを含むものである。図１０は、本編タイトル、オンラインショッピングタイトル、ゲームタイトルという3つのタイトルを含むディスクコンテンツを示す図である。本図における右側にはIndex Tableを記述しており、左側には3つのタイトルを記述している。

【００４２】

右側における破線枠は、各アプリケーションがどのタイトルに属しているかという帰属関係を示す。3つのタイトルのうちtitle#1は、application#1、application#2、application#3という3つのアプリケーションからなる。title#2は、application#3、application#4という2つのアプリケーション、title#3は、application#5を含む。図１１は、図１０に示した3つのタイトルの再生画像の一例を示す図である。これら3つのタイトルの再生画像において、図１１（ａ）（ｂ）の本編タイトル、オンラインショッピングタイトルには、ショッピングカートを模した映像(カートcr1)1が存在するが、図１１（ｃ）のゲームタイトルには、カート映像が存在しない。カートcr1は、本編タイトル、オンラインショッピングタイトルにおいて共通して表示しておく必要があるので、カートプログラムたるapplication#3を、title#1、title#2の双方で起動するようにしている。このように複数タイトルで起動するようなアプリケーションには、上述したカートアプリの他に、映画作品に登場するマスコットを模したエージェントアプリ、メニューコール操作に応じてメニュー表示を行うメニューアプリがある。

【００４３】

図１０の破線に示される帰属関係から各アプリケーションの生存区間をグラフ化すると、図１２（ａ）のようになる。本図において横軸は、タイトル時間軸であり、縦軸方向に各アプリケーションの生存区間を配置している。ここでapplication#1、application#2は、title#1のみに帰属しているので、これらの生存区間は、title#1内に留まっている。application#4は、title#2のみに帰属しているので、これらの生存区間は、title#2内に留まっている。application#5は、title#3のみに帰属しているので、これらの生存区間は、title#3内に留まっている。application#3は、title#1及びtitle#2に帰属しているので、これらの生存区間は、title#1－title#2にわたる。この生存区間に基づき、アプリケーション管理テーブルを記述すると、title#1,#2,#3のアプリケーション管理テーブルは図１２（ｂ）のようになる。このようにアプリケーション管理テーブルが記述されれば、title#1の再生開始時においてapplication#1、application#2、application#3をワークメモリにロードしておく。そしてtitle#2の開始時にapplication#1、application#2をワークメモリから削除してapplication#3のみにするという制御を行う。これと同様にtitle#2の再生開始時においてapplication#4をワークメモリにロードしておき、title#3の開始時にap

plication#3,#4をワークメモリから削除するという制御を行いうる。

【００４４】

更に、title#3の再生中においてapplication#5をワークメモリにロードしておき、title#3の再生終了時にapplication#5をワークメモリから削除するという制御を行いうる。

タイトル間分岐があった場合でも、分岐元ー分岐先において生存しているアプリケーションはワークメモリ上に格納しておき、分岐元にはなく、分岐先にのみ存在するアプリケーションをワークメモリに読み込めば良いから、アプリケーションをワークメモリに読み込む回数は必要最低数になる。このように、読込回数を少なくすることにより、タイトルの境界を意識させないアプリケーション、つまりアンバウンダリなアプリケーションを実現することができる。

【００４５】

続いてアプリケーションの起動属性について説明する。起動属性には、自動的な起動を示す「AutoRun」、自動起動の対象ではないが、仮想マシンのワークメモリに置いて良いことを示す「Persistent」、仮想マシンのワークメモリにはおかれるが、CPUパワーの割り当ては不可となる「Suspend」がある。

「AutoRun」は、対応するタイトルの分岐と同時に、そのアプリケーションをワークメモリに読み込み、且つ実行する旨を示す生存区間である。あるタイトルから、別のタイトルへの分岐があると、アプリケーション管理を行う管理主体(アプリケーションマネージャ)は、その分岐先タイトルにおいて生存しており、かつ起動属性がAutoRunに設定されたアプリケーションを仮想マシンのワークメモリに読み込み実行する。これによりそのアプリケーションは、タイトル分岐と共に自動的に起動されることになる。起動属性をAutoRunに設定しておくアプリケーションとしては、JMFプレーヤインスタンス及びJumpTitleAPIコールをもつようなアプリケーションが挙げられる。何故なら、このようなアプリケーションは、タイトル時間軸を律する側のアプリケーションであり、このようなアプリケーションを自動的に起動にしないと、タイトル時間軸の概念が曖昧になってしまうからである。

【００４６】

起動属性「Persistent」は、継続属性であり、分岐元titleにおけるアプリケーションの状態を継続することを示す。またワークメモリにロードしてよいことを示す属性である。起動属性が「Persistent」である場合、この起動属性が付与されたアプリケーションは、他のアプリケーションからの呼び出しが許可されることになる。アプリケーション管理を行う管理主体(アプリケーションマネージャ)は、起動中のアプリケーションから呼出があると、そのアプリケーションのapplicationIDが、アプリケーション管理テーブルに記述されていて、起動属性が「Persistent」であるか否かを判定する。「Persistent」であれば、そのアプリケーションをワークメモリにロードする。一方、その呼出先アプリケーションのapplicationIDがアプリケーション管理テーブルに記述されていない場合、そのアプリケーションはワークメモリにロードされない。アプリケーションによる呼出は、この「Persistent」が付与されたアプリケーションに限られることになる。

【００４７】

「Persistent」は、起動属性を明示的に指定しない場合に付与されるデフォルトの起動属性であるから、あるアプリケーションの起動属性が無指定「ーー」である場合、そのアプリケーションの起動属性の起動属性はこのPersistentであることを意味する。

これらの起動属性が、図１１のアプリケーションにおいてどのように記述されているかについて説明する。図１３は、図１２の3つのアプリケーションに対する起動属性の設定例である。図１２に示した3つのアプリケーションのうちapplication#2は、図１３（ｂ）に示すように他のアプリケーションからのアプリケーション呼出があって初めて起動するアプリケーションであるとする。残りのapplication#1、application#3は、title#1の開始と同時に自動的に起動されるアプリケーションであるとする。この場合、図１３（ａ）に示すように、アプリケーション管理テーブルにおける各アプリケーションの起動属性を、application#1、application#3は「AutoRun」、application#2は、「Persistent」と設

定しておく。この場合、application#1、application#3は、title#1への分岐時において自動的にワークメモリにロードされ、実行されることになる。一方application#2は、起動属性がPersistentなので、「application#2は仮想マシンのワークメモリ上にロードしてよいアプリケーション」であるとの消極的な意味に解される。故に、application#2は、application#1からの呼出があって初めて仮想マシンのワークメモリにロードされ、実行されることになる。以上の生存区間・起動属性により、仮想マシン上で動作し得るアプリケーションの数を4個以下に制限し、総スレッド数を64個以下に制限することが可能なので、アプリケーションの安定動作を保証することができる。

【００４８】

続いてSuspendについて説明する。

Suspendとは、リソースは割り付けられているが、CPUパワーは割り当てられない状態にアプリケーションが置かれることをいう。かかるSuspendは、例えばゲームタイトルの実行中に、サイドパスを経由するという処理の実現に有意義である。図１４（ａ）（ｂ）はSuspendが有意義となる事例を示す図である。図１４（ｂ）に示すように、3つのタイトル(title#1、title#2、title#3)があり、そのうちtitle#1、title#3はゲームアプリを実行するが、途中のtitle#2はサイドパスであり、映像再生を実現するものである。サイドパスでは、映像再生を実現する必要があるため、ゲームの実行を中断させることになる。ゲームアプリでは途中のスコア等が計数されているため、リソースの格納値はtitle#2の前後で維持したい。この場合、title#2の開始時点でゲームアプリをSuspendし、title#3の開始時点でapplication#2をレジュームするというようにアプリケーション管理テーブルを記述する。こうすることでtitle#2においてapplication#2は、リソースは割り付けられているので、リソースの格納値は維持される。しかし、CPUパワーは割り当てられない状態なので仮想マシンによりapplication#2は実行されることはない。これにより、ゲームタイトルの実行中に、サイドパスを実行するという処理が実現される。

【００４９】

図１５は、起動属性がとり得る三態様(Persistent、AutoRun、Suspend)と、直前タイトルにおけるアプリケーション状態の三態様(非起動、起動中、Suspend)とがとりうる組合せを示す図である。直前状態が”非起動”である場合、起動属性が”AutoRun”であるなら、分岐先タイトルにおいてそのアプリケーションは、起動されることになる。

直前状態が”非起動”であり、起動属性が”Persistent”、”Suspend”であるなら、分岐先タイトルにおいてそのアプリケーションは、何ももせず、状態を継続することになる。

【００５０】

直前状態が”起動中”である場合、起動属性が”Persistent”、”AutoRun”であるなら、分岐先タイトルにおいてそのアプリケーションは、何もせず、状態を継続することになる。

起動属性が”Suspend”であるなら、アプリケーションの状態はSuspendされることになる。直前状態が”Suspend”である場合、分岐先タイトルの起動属性が”Suspend”ならSuspendを維持することになる。”Persistent”、”AutoRun”であるなら、分岐先タイトルにおいてそのアプリケーションは、レジュームすることになる。アプリケーション管理テーブルにおいて生存区間及び起動属性を定義することにより、タイトル時間軸の進行に沿って、Javaアプリケーションを動作させるという同期制御が可能になり、映像再生と、プログラム実行とを伴った、様々なアプリケーションを世に送り出すことができる。以上が記録媒体についての説明である。続いて本発明に係る再生装置について説明する。

【００５１】

図１６は、本発明に係る再生装置の内部構成を示す図である。本発明に係る再生装置は、本図に示す内部に基づき、工業的に生産される。本発明に係る再生装置は、主としてシステムLSIと、ドライブ装置という2つのパーツからなり、これらのパーツを装置のキャビネット及び基板に実装することで工業的に生産することができる。システムLSIは、再生装置の機能を果たす様々な処理部を集積した集積回路である。こうして生産される再生装

置は、BD-ROMドライブ１、リードバッファ２、デマルチプレクサ３、ビデオデコーダ４、ビデオプレーン５、P-Graphicsデコーダ９、Presentation Graphicsプレーン１０、合成部１１、フォントゼネレータ１２、I-Graphicsデコーダ１３、スイッチ１４、Interactive Graphicsプレーン１５、合成部１６、HDD１７、リードバッファ１８、デマルチプレクサ１９、オーディオデコーダ２０、シナリオメモリ２１、CPU２２、キーイベント処理部２３、命令ROM２４、スイッチ２５、CLUT部２６、CLUT部２７、PSRセット２８、ローカルメモリ２９から構成される。

【００５２】

BD-ROMドライブ１は、BD-ROMのローディング／イジェクトを行い、BD-ROMに対するアクセスを実行する。

リードバッファ２は、FIFOメモリであり、BD-ROMから読み出されたTSパケットが先入れ先出し式に格納される。

デマルチプレクサ(De-MUX)３は、リードバッファ２からTSパケットを取り出して、このTSパケットを構成するTSパケットをPESパケットに変換する。そして変換により得られたPESパケットのうち、CPU２２から設定されたPIDをもつものをビデオデコーダ４、オーディオデコーダ２０、P-Graphicsデコーダ９、I-Graphicsデコーダ１３のどれかに出力する。

【００５３】

ビデオデコーダ４は、デマルチプレクサ３から出力された複数PESパケットを復号して非圧縮形式のピクチャを得てビデオプレーン５に書き込む。

ビデオプレーン５は、非圧縮形式のピクチャを格納しておくためのプレーンである。プレーンとは、再生装置において一画面分の画素データを格納しておくためのメモリ領域である。再生装置に複数のプレーンを設けておき、これらプレーンの格納内容を画素毎に加算して、映像出力を行えば、複数の映像内容を合成させた上で映像出力を行うことができる。ビデオプレーン５における解像度は1920×1080であり、このビデオプレーン５に格納されたピクチャデータは、16ビットのYUV値で表現された画素データにより構成される。

【００５４】

P-Graphicsデコーダ９は、BD-ROM、HDD１７から読み出されたプレゼンテーショングラフィクスストリームをデコードして、非圧縮グラフィクスをPresentation Graphicsプレーン１０に書き込む。グラフィクスストリームのデコードにより、字幕が画面上に現れることになる。

Presentation Graphicsプレーン１０は、一画面分の領域をもったメモリであり、一画面分の非圧縮グラフィクスを格納することができる。本プレーンにおける解像度は1920×1080であり、Presentation Graphicsプレーン１０中の非圧縮グラフィクスの各画素は8ビットのインデックスカラーで表現される。CLUT(Color Lookup Table)を用いてかかるインデックスカラーを変換することにより、Presentation Graphicsプレーン１０に格納された非圧縮グラフィクスは、表示に供される。

【００５５】

合成部１１は、非圧縮状態のピクチャデータ(i)を、Presentation Graphicsプレーン１０の格納内容と合成する。

フォントゼネレータ１２は、文字フォントを用いてtextSTストリームに含まれるテキストコードをビットマップに展開する。

I-Graphicsデコーダ１３は、BD-ROM又はHDD１７から読み出されたインタラクティブグラフィクスストリームをデコードして、非圧縮グラフィクスをInteractive Graphicsプレーン１５に書き込む。

【００５６】

スイッチ１４は、フォントゼネレータ１２が生成したフォント列、P-Graphicsデコーダ９のデコードにより得られたグラフィクスの何れかを選択的にPresentation Graphicsプレーン１０に書き込むスイッチである。

Interactive Graphicsプレーン１５は、I-Graphicsデコーダ１３によるデコードで得られた非圧縮グラフィクスが書き込まれる。

【００５７】
合成部１６は、Interactive Graphicsプレーン１０の格納内容と、合成部８の出力である合成画像(非圧縮状態のピクチャデータと、Presentation Graphicsプレーン７の格納内容とを合成したもの)とを合成する。
HDD１７は、ネットワーク等を介してダウンロードされたSubClip、Clip情報、プレイリスト情報が格納される内蔵媒体である。このHDD１７中のプレイリスト情報はBD-ROM及びHDD１７のどちらに存在するClip情報であっても、指定できる点で異なる。この指定にあたって、HDD１７上のプレイリスト情報は、BD-ROM上のファイルをフルパスで指定する必要はない。本HDD１７は、BD-ROMと一体になって、仮想的な1つのドライブ(バーチャルパッケージと呼ばれる)として、再生装置により認識されるからである。故に、PlayItem情報におけるClip_Information_file_name及びSubPlayItem情報のClip_Information_file_nameは、Clip情報の格納したファイルのファイルボデイにあたる5桁の数値を指定することにより、HDD１７、BD-ROM上のAVClipを指定することができる。このHDDの記録内容を読み出し、BD-ROMの記録内容と動的に組み合わせることにより、様々な再生のバリエーションを産み出すことができる。
【００５８】
リードバッファ１８は、FIFOメモリであり、HDD１７から読み出されたTSパケットが先入れ先出し式に格納される。
デマルチプレクサ(De-MUX)１９は、リードバッファ１８からTSパケットを取り出して、TSパケットをPESパケットに変換する。そして変換により得られたPESパケットのうち、所望のstreamPIDをもつものをフォントゼネレータ１２に出力する。
【００５９】
オーディオデコーダ２０は、デマルチプレクサ１９から出力されたPESパケットを復号して、非圧縮形式のオーディオデータを出力する。
シナリオメモリ２１は、カレントのPL情報やカレントのClip情報を格納しておくためのメモリである。カレントPL情報とは、BD-ROMに記録されている複数PL情報のうち、現在処理対象になっているものをいう。カレントClip情報とは、BD-ROMに記録されている複数Clip情報のうち、現在処理対象になっているものをいう。
【００６０】
CPU２２は、命令ROM２４に格納されているソフトウェアを実行して、再生装置全体の制御を実行する。
キーイベント処理部２３は、リモコンや再生装置のフロントパネルに対するキー操作に応じて、その操作を行うキーイベントを出力する。
命令ROM２４は、再生装置の制御を規定するソフトウェアを記憶している。
【００６１】
スイッチ２５は、BD-ROM及びHDD１７から読み出された各種データを、リードバッファ２、リードバッファ１８、シナリオメモリ２１、ローカルメモリ２９のどれかに選択的に投入するスイッチである。
CLUT部２６は、ビデオプレーン５に格納された非圧縮グラフィクスにおけるインデックスカラーを、Y,Cr,Cb値に変換する。
【００６２】
CLUT部２７は、Interactive Graphicsプレーン１５に格納された非圧縮グラフィクスにおけるインデックスカラーを、Y,Cr,Cb値に変換する。
PSRセット２８は、再生装置に内蔵されるレジスタであり、64個のPlayer Status Register(PSR)と、4096個のGeneral Purpose Register (GPR)とからなる。Player Status Registerの設定値(PSR)のうち、PSR4～PSR8は、現在の再生時点を表現するのに用いられる。
【００６３】
PSR4は、1～100の値に設定されることで、現在の再生時点が属するタイトルを示し、0に設定されることで、現在の再生時点がトップメニューであることを示す。
PSR5は、1～999の値に設定されることで、現在の再生時点が属するチャプター番号を示

し、0xFFFFに設定されることで、再生装置においてチャプター番号が無効であることを示す。

【００６４】

PSR6は、0～999の値に設定されることで、現在の再生時点が属するPL(カレントPL)の番号を示す。

PSR7は、0～255の値に設定されることで、現在の再生時点が属するPlay Item(カレントPlay Item)の番号を示す。

PSR8は、0～0xFFFFFFFFの値に設定されることで、45KHzの時間精度を用いて現在の再生時点(カレントPTM(Presentation TiMe))を示す。以上のPSR4～PSR8により、現在の再生時点が特定されることになる。

【００６５】

ローカルメモリ２９は、BD-ROMからの読み出しは低速である故、BD-ROMの記録内容を一時的に格納しておくためのキャッシュメモリである。かかるローカルメモリ２９が存在することにより、BD-Jモードにおけるアプリケーション実行は、効率化されることになる。図１７（ａ）は、BD-ROMに存在しているJavaアーカイブファイルを、ローカルメモリ２９上でどのように識別するかを示す図である。図１７（ａ）の表は、左欄にBD-ROM上のファイル名を、右欄にローカルメモリ２９上のファイル名をそれぞれ示している。これら右欄、左欄を比較すれば、ローカルメモリ２９におけるファイルは、ディレクトリ指定"BDJA"を省いたファイルパスで指定されていることがわかる。

【００６６】

図１７（ｂ）は、図１７（ａ）の応用を示す図である。本応用例は、ヘッダ＋データという形式で、ファイルに格納されているデータを格納するというものである。何をヘッダに用いるかというと、ローカルメモリ２９におけるファイルパスを用いる。図１７（ｂ）に示したように、ローカルメモリ２９ではBD-ROMにおけるファイルパスの一部を省略したものをファイルパスに用いるから、当該ファイルパスをヘッダに格納することで、各データにおけるBD-ROM上の所在を明らかにすることができる。

【００６７】

以上が、本実施形態に係る再生装置のハードウェア構成である。続いて本実施形態に係る再生装置のソフトウェア構成について説明する。

図１８は、ROM２４に格納されたソフトウェアと、ハードウェアとからなる部分を、レイア構成に置き換えて描いた図である。本図に示すように、再生装置のレイア構成は、以下のa),b),c),d-1),d-2),e),f)からなる。つまり、

a)物理的なハードウェア階層の上に、

b)AVClipによる再生を制御するPresentation Engine３１、

c)プレイリスト情報及びClip情報に基づく再生制御を行うPlayback Control Engine３２、

という2つの階層があり、

最上位の階層に

e)タイトル間の分岐を実行するモジュールマネージャ３４がある。

【００６８】

これらHDMVモジュール３３、モジュールマネージャ３４の間に、

d-1)Movieオブジェクトの解読・実行主体であるHDMVモジュール３３と、

d-2)BD-Jオブジェクトの解読・実行を行うBD-Jモジュール３５とが同じ階層に置かれている。

BD-Jモジュール３５は、いわゆるJavaプラットフォームであり、ワークメモリ３７を含むJava仮想マシン３８を中核にした構成になっていて、アプリケーションマネージャ３６、Event Listner Manager３９、Default Operation Manager４０から構成される。先ず初めに、Presentation Engine３１～モジュールマネージャ３４について説明する。図１９は、Presentation Engine３１～モジュールマネージャ３４による処理を模式化した図である。

【００６９】

Presentation Engine３１は、AV再生機能を実行する。再生装置のAV再生機能とは、DVDプレーヤ、CDプレーヤから踏襲した伝統的な機能群であり、再生開始(Play)、再生停止(Stop)、一時停止(Pause On)、一時停止の解除(Pause Off)、Still機能の解除(still off)、速度指定付きの早送り(Forward Play(speed))、速度指定付きの巻戻し(Backward Play(speed))、音声切り換え(Audio Change)、副映像切り換え(Subtitle Change)、アングル切り換え(Angle Change)といった機能である。AV再生機能を実現するべく、Presentation Engine３１は、リードバッファ２上に読み出されたAVClipのうち、所望に時刻にあたる部分のデコードを行うよう、ビデオデコーダ４、P-Graphicsデコーダ９、I-Graphicsデコーダ１３、オーディオデコーダ２０を制御する。所望の時刻としてPSR8(カレントPTM)に示される箇所のデコードを行わせることにより、AVClipにおいて、任意の時点を再生を可能することができる。図中の1は、Presentation Engine３１によるデコード開始を模式化して示す。

【００７０】

再生制御エンジン(Playback Control Engine(PCE))３２は、プレイリストの再生機能(i)、再生装置における状態取得／設定機能(ii)といった諸機能を実行する。PLの再生機能とは、Presentation Engine３１が行うAV再生機能のうち、再生開始や再生停止を、カレントPL情報及びClip情報に従って行わせることをいう。これら機能(i)～(ii)は、HDMVモジュール３３～BD-Jモジュール３５からのファンクションコールに応じて実行する。つまり再生制御エンジン３２は、ユーザ操作による指示、レイヤモデルにおける上位層からの指示に応じて、自身の機能を実行する。図１９において、2,3が付された矢印は、Clip情報及びプレイリスト情報に対するPlayback Control Engine３２の参照を模式的に示す。

【００７１】

HDMVモジュール３３は、MOVIEモードの実行主体であり、モジュールマネージャ３４から分岐先を構成するMovieオブジェクトが通知されれば、分岐先タイトルを構成するMovieオブジェクトをローカルメモリ２９に読み出して、このMovieオブジェクトに記述されたナビゲーションコマンドを解読し、解読結果に基づきPlayback Control Engine３２に対するファンクションコールを実行する。図１９において▽2,▽3,▽4が付された矢印は、モジュールマネージャ３４からの分岐先Movieオブジェクトの通知(2)、Movieオブジェクトに記述されたナビゲーションコマンドの解読(3)、Playback Control Engine３２に対するファンクションコール(4)を、模式的に示している。

【００７２】

モジュールマネージャ３４は、BD-ROMから読み出されたIndex Tableを保持して、分岐制御を行う。この分岐制御は、JumpTitleコマンドをHDMVモジュール３３が実行した場合、又は、タイトルジャンプAPIがBD-Jモジュール３５からコールされた場合、そのジャンプ先となるタイトル番号を受け取って、そのタイトルを構成するMovieオブジェクト又はBD-JオブジェクトをHDMVモジュール３３又はBD-Jモジュール３５に通知するというものである。図中の▽0,▽1,▽2が付された矢印は、JumpTitleコマンドの実行(0)、モジュールマネージャ３４によるIndexTable参照(1)、分岐先Movieオブジェクト(2)の通知を模式的に示している。

【００７３】

以上がPresentation Engine３１～モジュールマネージャ３４についての説明である。続いてアプリケーションマネージャ３６について、図２０を参照しながら説明する。図２０は、アプリケーションマネージャ３６を示す図である。

アプリケーションマネージャ３６は、アプリケーション管理テーブルを参照したアプリケーションの起動制御、タイトルの正常終了時における制御を実行する。

【００７４】

起動制御とは、モジュールマネージャ３４から分岐先となるBD-Jオブジェクトが通知される度に、そのBD-Jオブジェクトを読み出し、そのBD-Jオブジェクト内のアプリケーション管理テーブルを参照してローカルメモリ２９アクセスを行う。そして現在の再生時点を

生存区間とするアプリケーションを構成するxletプログラムを、ワークメモリに読み出すという制御である。図２０における☆1,☆2,☆3は、起動制御における分岐先BD-Jオブジェクトの通知(1)、アプリケーション管理テーブル参照(2)、Java仮想マシン３８に対する起動指示を模式化して示す。この起動指示によりJava仮想マシン３８は、ローカルメモリ２９からワークメモリ３７にxletプログラムを読み出す(☆5)。

【００７５】

タイトルの終了制御には、正常終了時の制御と、異常終了時の制御とがある。正常終了時の制御には、タイトルを構成するアプリケーションによりジャンプタイトルAPIがコールされて、分岐先タイトルへの切り換えを分岐制御の主体(モジュールマネージャ３４)に要求するという制御がある。この終了制御における、モジュールマネージャ３４通知を模式化して示したのが矢印☆6である。ここでタイトルを正常終了するにあたって、タイトルを構成するアプリケーションは、起動されたままであってもよい。何故なら、アプリケーションを終了するか否かは、分岐先タイトルにおいて判断するからである。本実施形態では深く触れないが、アプリケーションマネージャ３６は、BD-ROMからローカルメモリ２９にJavaアーカイブファイルを読み出す(8)との処理を行う。このローカルメモリ２９への読み出しを模式化したのが☆8である。

【００７６】

以上がアプリケーションマネージャ３６についての説明である。続いてワークメモリ３７～Default Operation Manager４０について、図２１を参照しながら説明する。

ワークメモリ３７は、アプリケーションを構成するxletプログラムが配置されるヒープ領域である。ワークメモリ３７は、本来Java仮想マシン３８内に存在するが、図２１では、作図の便宜上、ワークメモリ３７をJava仮想マシン３８上位層に記述している。ワークメモリ３７上のxletプログラムには、EventListnerや、JMFプレーヤインスタンスが含まれる。

【００７７】

Java仮想マシン３８は、アプリケーションを構成するxletプログラムをワークメモリ３７にロードして、xletプログラムを解読し、解読結果に従った処理を実行する。上述したようにxletプログラムは、JMFプレーヤインスタンス生成を命じるメソッド、このJMFプレーヤインスタンスの実行を命じるメソッドを含むので、これらのメソッドで命じられた処理内容を実現するよう、下位層に対する制御を行う。JMFプレーヤインスタンス生成が命じられれば、Java仮想マシン３８は、BD-ROM上のYYYY.MPLSファイルに関連付けられたJMFプレーヤインスタンスを得る。また、JMFプレーヤインスタンスにおけるJMFメソッドの実行が命じられれば、このJMFメソッドをBDミドルウェアに発行して、BD再生装置が対応しているファンクションコールに置き換させる。そして置換後のファンクションコールをPlayback Control Engine３２に発行する。

【００７８】

Event Listner Manager３９は、ユーザ操作により生じたイベント(キーイベント)を解析し、イベントの振り分けを行う。図中の実線矢印◇1,◇2は、このEvent Listner Manager３９による振り分けを模式的に示す。START、STOP、SPEED等、xletプログラム内のEvent Listnerに登録されたキーイベントなら、BD-Jオブジェクトにより間接参照されているxletプログラムにかかるイベントを振り分ける。START、STOP、SPEEDは、JMFに対応したイベントであり、xletプログラムのEvent Listnerには、これらのキーイベントが登録されているので、本キーイベントによりxletプログラムの起動が可能になる。キーイベントがEvent Listner非登録のキーイベントである場合、本キーイベントをDefault Operation Manager４０に振り分ける。音声切り換え、アングル切り換え等、BD-ROM再生装置において生じるキーイベントには、Event Listnerに登録されていない多様なものがあり、これらのキーイベントが生じたとしても、漏れの無い処理を実行するためである。

【００７９】

Default Operation Manager４０は、xletプログラム内のEvent Listnerに登録されてないキーイベントがEvent Listner Manager３９から振り分けられると、そのEvent Listner

非登録イベントに対応するファンクションコールをPlayback Control Engine３２に対して実行する。このDefault Operation Manager４０によるファンクションコールを模式的に示したのが、図中の矢印◇3である。尚、図２１においてEvent Listner非登録イベントはEvent Listner Manager３９、Default Operation Manager４０により振り分けられたが、Playback Control Engine３２がダイレクトにEvent Listner非登録イベントを受け取り、再生制御を行ってもよい(図中の◇4)。

。

【００８０】

(フローチャートの説明)

以上のアプリケーションマネージャ３６についての説明は、その概要に触れたに過ぎない。アプリケーションマネージャ３６の処理を更に詳しく示したのが図２２、図２３のフローチャートである。以降、これらのフローチャートを参照してアプリケーションマネージャ３６の処理手順についてより詳しく説明する。

【００８１】

図２２は、アプリケーションマネージャ３６による分岐時の制御手順を示す図である。本フローチャートは、ステップＳ２～ステップＳ５がなす条件を満たすアプリケーション(アプリケーションxという)を、起動又は終了させるという処理である。

ステップＳ２は、分岐元タイトルで非起動だが、分岐先タイトルで生存していて、分岐先タイトルにおける起動属性がAutoRun属性のアプリケーションxが存在するか否かの判定であり、もしあれば、ローカルメモリ２９に対するキャッシュセンスを行う。キャッシュセンスの結果、アプリケーションxがローカルメモリ２９上に有れば(ステップＳ７でYes)、ローカルメモリ２９からワークメモリ３７にアプリケーションxを読み込む(ステップＳ８)。ローカルメモリ２９に無ければ、BD-ROMからローカルメモリ２９にアプリケーションxを読み込んだ上で、ローカルメモリ２９からワークメモリ３７にアプリケーションxを読み込む(ステップＳ９)。

【００８２】

ステップＳ３は、分岐元タイトルで起動中で、分岐先タイトルで非生存のアプリケーションxが存在するかどうかの判定である。もし存在するのなら、アプリケーションxをワークメモリ３７から削除して終了させる(ステップＳ１０)。

ステップＳ４は、分岐元Suspend、分岐先AutoRun又はPersistentのアプリケーションが存在するか否かの判定である。もし存在するなら、アプリケーションxをResumeする(ステップＳ１１)。

【００８３】

ステップＳ５は、分岐元タイトルで起動中で、分岐先Suspendのアプリケーションが存在するか否かの判定である。もし存在すれば、アプリケーションxをSuspendする(ステップＳ１２)。

個々のアプリケーションを終了させるにあたってのアプリケーションマネージャ３６の処理は、図２３に示すものとなる。図２３は、アプリケーション終了処理の処理手順を示すフローチャートである。本図は、終了すべき複数アプリケーションのそれぞれについて、ステップＳ１６～ステップＳ２０の処理を繰り返すループ処理になっている(ステップＳ１５)。本ループ処理においてアプリケーションマネージャ３６は起動中アプリケーションを終了するようなterminateイベントを発行し(ステップＳ１６)、タイマセットして(ステップＳ１７)、ステップＳ１８～ステップＳ２０からなるループ処理に移行する。このterminateイベントをEvent Listnerが受信すれば、対応するxletプログラムは、終了プロセスを起動する。終了プロセスが終了すれば、そのxletプログラムはワークメモリ３７から解放され、終了することになる。

【００８４】

ステップＳ１８～ステップＳ２０のループ処理の継続中、タイマはカウントダウンを継続する。本ループ処理においてステップＳ１８は、発行先アプリケーションが終了したか否かの判定であり、もし終了していればこのアプリケーションに対する処理を終える。ス

テップＳ１９は、タイマがタイムアウトしたか否かの判定であり、タイムアウトすれば、ステップＳ２０において発行先アプリケーションをワークメモリ３７から削除してアプリケーションを強制終了する。

【００８５】

以上のモジュールマネージャ３４の処理を、図２４を参照しながら説明する。

図２４は、アプリケーション終了の過程を模式的に示した図である。本図における第1段目は、アプリケーションマネージャ３６を、第2段目は、3つのアプリケーションを示す。図２４の第2段目、左側のアプリケーションは、terminateイベントを受信して終了プロセスに成功したアプリケーションを示す。図２４の第2段目、中列のアプリケーションは、terminateイベントを受信したが終了プロセスに失敗したアプリケーションを示す。第2段目、右側のアプリケーションは、EventListnerが実装されていないので、terminateイベントを受信することができなかったアプリケーションを示す。

【００８６】

第1段目－第2段目間の矢印ep1,ep2は、アプリケーションマネージャによるterminateイベント発行を模式的に示し、矢印ep3は、終了プロセス起動を模式的に示している。

第3段目は、終了プロセス成功時における状態遷移後の状態であり、このアプリケーションは、自身の終了プロセスにより終了することになる。これらxletプログラムのように、所定の期間内に終了しないアプリケーションがあれば、アプリケーションマネージャ３６は、それらを強制的にワークメモリ３７から取り除く。第4段目は、アプリケーションマネージャ３６による強制終了を示す。かかる第4段目の強制終了を規定するのも、アプリケーションマネージャ３６の1つの使命といえる。

【００８７】

以上のように本実施形態によれば、分岐元タイトルで起動しており、分岐先タイトルで生存していないアプリケーションは、自動的に終了させられるので、条件付き分岐により再生が複雑に進行する場合でも、再生装置におけるリソースの限界を越える数のアプリケーション立ち上げはなされ無い。分岐前後におけるアプリケーション動作を保証することができるので、デジタルストリームを再生させながら、アプリケーションを実行させるようなディスクコンテンツを多く頒布することができる。

【００８８】

（第２実施形態）

第１実施形態においてアプリケーションの生存区間は、タイトル時間軸と一致していたが、第２実施形態は、PL時間軸の一部をアプリケーションの生存区間とすることを提案する。PL時間軸の一部は、チャプターにより表現されるので、チャプターにて開始点、終了点を記述することにより、アプリケーションの生存区間を規定することができる。図２５（ａ）は、PL時間軸上に生存区間を定めたアプリケーション管理テーブルを示す図である。図２５（ａ）においてアプリケーション管理テーブルには、3つのアプリケーションが記述されており、このうちapplication#2は、title#1のChapter#2からChapter#3までが生存区間に指定され、起動属性にAutoRunが規定されている。このためapplication#2は、図２５（ｂ）に示すように、Chapter#2の始点で起動され、Chapter#3の終点で終了することになる。

【００８９】

一方application#3は、title#1のChapter#4からChapter#6までが生存区間に指定されている。このためapplication#3は、図２５（ｂ）に示すように、図２５（ｂ）に示すように、Chapter#4の始点で起動され、Chapter#6の終点で終了することになる。

こうして記述されたアプリケーション管理テーブルに基づき、処理を行うため本実施形態に係るアプリケーションマネージャ３６は、PLmarkにより指示されるチャプター開始点に到達する度に、そのチャプター開始点から生存区間が始まるアプリケーションが存在するか否かを判定し、もし存在すればそのアプリケーションをワークメモリ３７にロードする。

【００９０】

同様に、チャプター開始点に到達する度に、そのチャプターの直前のチャプターで生存区間が終わるアプリケーションが存在するか否かを判定し、もし存在すればそのアプリケーションをワークメモリ３７から解放する。

チャプターという単位でアプリケーションの生存を管理すれば、アプリケーションの生存区間をより細かい精度で指定することができる。しかしディスクコンテンツには、時間軸の逆向がありうることに留意せねばならない。逆行とは、巻戻しにより時間軸を逆向きに進行することである。チャプターの境界でこの逆行と、進行とが繰り返されれば、ワークメモリへのロード、廃棄が何度もなされ、余分な読出負荷が生じる。そこで本実施形態では、アプリケーションの起動時期を、タイトルに入ってPlayback Control Engine３２による通常再生が開始された瞬間にしている。ここでPLの再生には、通常再生、トリック再生がある。トリック再生とは、早送り、巻戻し、SkipNext,SkipBackがある。かかる、早送り、巻戻し、SkipNext,SkipBackがなされている間、アプリケーション起動を開始せず、通常再生が開始されて初めて、アプリケーションを起動するのである。通常再生開始の瞬間を基準にすることにより、上述したような生存区間前後の行き来があった場合でも、アプリケーションの起動が必要以上に繰り返されることはない。尚、通常再生開始の瞬間を、アプリケーションの起動基準にするとの処理は、生存区間がtitleである場合にも、実行してよい。

【００９１】

以上のように本実施形態によれば、PLより小さい、チャプターの単位でアプリケーションの生存区間を規定することができるので、緻密なアプリケーション制御を実現することができる。

（第２実施形態の変更例）

図２５では、各アプリケーションに優先度が付与されている。この優先度は、0～255の値をとり、アプリケーション間でリソースの使用が競合等が競合した場合、どちらのアプリケーションを強制的に終了させるか、また、どちらのアプリケーションからリソースを奪うかという処理をアプリケーションマネージャ３６が行うにあたって、判断材料になる。図２５の一例では、application#1の優先度は255,application#2、application#3の優先度は128なので、application#1－application#2の競合時において、アプリケーションマネージャ３６は優先度が低いapplication#2を強制終了するとの処理を行う。

【００９２】

（第３実施形態）

BD-ROMにより供されるディスクコンテンツは、互いに分岐可能な複数タイトルから構成される。各タイトルは、1つ以上のPLと、このPLを用いた制御手順とからなるもの以外に、再生装置に対する制御手順のみからなる非AV系タイトルがある。本実施形態は、この非AV系タイトルについて説明する。

【００９３】

こうした非AV系タイトルのタイトルでは、どのようにタイトル時間軸を定めるのかが問題になる。図２６（ａ）は、PL時間軸から定まるタイトル時間軸を示す。この場合PL時間軸がタイトル時間軸になり、このタイトル時間軸上にアプリケーションの生存区間が定まる。この基準となるPL時間軸がない場合、タイトル時間軸は図２６（ｂ）（ｃ）のように定めるべきである。

【００９４】

図２６（ｂ）は、メインとなるアプリケーションの生存区間から定まるタイトル時間軸を示す。メインアプリとは、タイトルにおいて起動属性がAutoRunに設定され、タイトル開始時に自動起動される唯一のアプリケーションであり、例えばランチャーアプリと呼ばれるものがこれにあたる。ランチャーアプリとは、他のアプリケーションを起動するアプリケーションプログラムである。

【００９５】

この図２６（ｂ）の考え方は、メインアプリが起動している限り、タイトル時間軸は継続していると考え、メインアプリが終了すれば、時間軸を終結させるというものである。

図２６（ｃ）は、複数アプリケーションの生存区間から定まるタイトル時間軸を示す図である。タイトルの開始点で起動されるのは1つのアプリケーションであるが、このアプリケーションが他のアプリケーションを呼び出し、更にこのアプリケーションが別のアプリケーションを呼び出すとの処理が繰り返されるというケースがある。この場合、どれかのアプリケーションが起動している限り、タイトル時間軸は継続していると考え、どのアプリケーションも起動していない状態が到来すれば、そこでタイトル時間軸は終結するという考え方である。このように非AV系タイトルのタイトル時間軸を定めれば、AVタイトルであっても、非AV系タイトルであっても、タイトル時間軸の終結と同時に、所定のタイトルに分岐するという処理を画一的に行うことができる。尚非AVタイトルにおけるタイトル時間軸は、AVタイトルと対比する上で、想定した架空の時間軸に過ぎない。故に再生装置は、非AVタイトルにおけるタイトル時間軸上を逆行したり、任意の位置に頭出しすることができない。

【００９６】

以上は本実施形態における記録媒体に対する改良である。続いて本実施形態における再生装置に対する改良について説明する。

上述したような手順でタイトル終了を行うため第３実施形態に係るアプリケーションマネージャ３６は、図２７に示すような処理で処理を行う。図２７は、タイトル再生時におけるアプリケーションマネージャ３６の処理手順を示すフローチャートである。本フローチャートは、タイトル再生中、ステップＳ２１～ステップＳ２３を繰り返すというループ構造になっている。

【００９７】

ステップＳ２１は、タイトルジャンプAPIが呼び出されたか否かの判定であり、呼び出されれば、ジャンプ先タイトルへの分岐をモジュールマネージャ３４に要求する(ステップＳ２７)。

ステップＳ２２は、タイトル内のアプリケーション呼出を担っているようなメインアプリが存在するか否かの判定であり、もし存在するなら、それの起動の有無を確認する(ステップＳ２５)。起動してなければ、”タイトルの終わり”であると解釈し、モジュールマネージャ３４に終結を通知する(ステップＳ２６)。

【００９８】

ステップＳ２３は、メインアプリがない場合に実行されるステップであり(ステップＳ２２でNo)、どのアプリケーションも起動してない状態かどうかを判定する。もしそうなら、同じく”タイトルの終わり”であると解釈し、モジュールマネージャ３４に終結を通知する(ステップＳ２６)。

以上のように本実施形態によれば、PL再生を伴わないタイトルであっとしても、アプリケーション実行中は分岐せず、アプリケーション実行が終了して初めて分岐するという処理が可能になる。

【００９９】

(第４実施形態)

本実施形態は、DVDと同様のメニュー制御をBD-ROM上で実現する場合の改良に関する。図２８（ａ）は、BD-ROMにより実現されるメニュー階層を示す図である。本図におけるメニュー階層は、TopMenuを最上位に配し、このTopMenuから下位のTitleMenu、SubTitleMenu、AudioMenuを選択できる構造になっている。図中の矢印sw1,2,3は、ボタン選択によるメニュー切り換えを模式的に示す。TopMenuとは、音声選択、字幕選択、タイトル選択の何れを行うかを受け付けるボタン(図中のボタンsn1,sn2,sn3)を配置したメニューである。

【０１００】

TitleMenuとは、映画作品(title)の劇場版を選択するか、ディレクターズカット版を選択するか、ゲーム版を選択するか等、映画作品の選択を受け付けるボタンを配置したメニューである。AudioMenuとは、音声再生を日本語で行うか、英語で行うかを受け付けるボタンを配置したメニュー、SubTitleMenuとは、字幕表示を日本語で行うか、英語で行うか

を受け付けるボタンを配置したメニューである。

【０１０１】

こうした階層をもったメニューを動作させるためのMOVIEオブジェクトを図２８（ｂ）に示す。図２８（ｂ）においてMovieObject.bdmvには、FirstPlay OBJ、TopMenu OBJ、AudioMenu OBJ、SubTitleMenu OBJが格納されている。

FirstPlayオブジェクト(FirstPlay OBJ)は、再生装置へのBD-ROMのローディング時に自動的に実行される動的シナリオである。

【０１０２】

TopMenuオブジェクト(TopMenu OBJ)は、TopMenuの挙動を制御する動的シナリオである。ユーザがメニューコールを要求した際、呼び出されるのはこのTopMenuオブジェクトである。TopMenuオブジェクトは、ユーザからの操作に応じてTopMenu中のボタンの状態を変えるものや、ボタンに対する確定操作に応じて分岐を行う分岐コマンドを含む。この分岐コマンドは、TopMenuからTitleMenu、TopMenuからSubTitleMenu、TopMenuからAudioMenuというメニュー切り換えを実現するものである。

【０１０３】

AudioMenuオブジェクト(AudioMenu OBJ)は、AudioMenuの挙動を制御する動的シナリオであり、ユーザからの操作に応じてAudioMenu中のボタンの状態を変えるコマンドや、ボタンに対する確定操作に応じて音声設定を更新するコマンドを含む。

SubTitleMenuオブジェクト(SubTitleMenu OBJ)は、SubTitleMenuの挙動を制御する動的なシナリオであり、ユーザからの操作に応じてSubTitleMenu中のボタンの状態を変えるコマンドや、ボタンに対する確定操作に応じて字幕設定用のPSRを更新するコマンドを含む。

【０１０４】

TitleMenuオブジェクト(TitleMenu OBJ)は、TitleMenuの挙動を制御する動的シナリオであり、TitleMenu中のボタンの状態を変えるものや、ボタンに対する確定操作に応じて分岐を行う分岐コマンドをふくむ。

これらのメニュー用MOVIEオブジェクトにより、DVDで実現されているようなメニューの挙動を実現することができる。以上がメニュー制御に関連するMOVIEオブジェクトである。

【０１０５】

図２９は、Index Tableと、Index Tableから各Movieオブジェクトへの分岐とを模式化した図である。本図では左側にIndex Tableの内部構成を示している。本実施形態におけるIndex Tableには、FirstPLayINDEX、TopMenuINDEX, Audio MenuINDEX、Subtitle MenuINDEX、title MenuINDEX、title#1～#mINDEX、title#m+1～#nINDEX、title#0INDEXを含む。図中の矢印bc1,2は、Index TableからFirstPlayOBJへの分岐と,FirstPlayOBJからTopMenuへの分岐とを模式的に示し、矢印bc3,4,5は、TopMenuからTitleMenu、SubTitleMenu、AudioMenuへの分岐を模式的に示している。矢印bc6,7,8は、TitleMenuから各Movieオブジェクトへの分岐を模式的に示している。

【０１０６】

FirstPLayINDEX、TopMenuINDEX, Audio MenuINDEX、Subtitle MenuINDEX、title MenuINDEXは、それぞれ、FirstPLayOBJ、TopMenuOBJ, Audio MenuOBJ、Subtitle MenuOBJ、title MenuOBJについてのIndexであり、これらの識別子が記述される。

title#1～#mINDEXは、BD-ROMにおいて1からm番目にエントリーされているtitleについてのIndexであり、これら1からmまでのtitle番号の選択時において分岐先となるMOVIEオブジェクトの識別子(ID)が記述される。

【０１０７】

title#m+1～#nINDEXは、BD-ROMにおいてm+1からn番目にエントリーされているtitleについてのIndexであり、これらm+1からnまでのtitle番号の選択時において分岐先となるBD-Jオブジェクトの識別子(ID)が記述される。

title#0INDEXは、BD-Jオブジェクトの強制終了時において分岐先になるべきMovieオブ

ジェクト又はBD-Jオブジェクトを規定するINDEXである。本実施形態では、TopMenuOBJについての識別子が、このtitle#0INDEXに格納されている。

【０１０８】

図３０（ａ）は、図２９のようにIndex Tableが記述された場合における分岐を示す。Index Tableがこのように記述されているので、ラベルtitle#1～title#mを分岐先とした分岐コマンドの実行時には、title#1Index～title#mIndexからMovieオブジェクト#1～#mの識別子が取り出される。ラベルtitle#m+1～title#nを分岐とした分岐コマンドの実行時には、title#m+1Index～title#nIndexからBD-Jオブジェクト#m+1～#nの識別子が取り出される。BD-Jオブジェクト#m+1～#nの識別子は、ファイル名を表す5桁の数値であるので、『00001.BD-J,00002.BD-J,00003.BD-J・・・』が取り出され、そのファイル名の動的シナリオがメモリに読み出されて、実行されることになる。これがIndex Tableを用いた分岐処理である。

【０１０９】

図３０（ｂ）は、BD-Jオブジェクト実行時の強制終了時における分岐を示す図である。強制終了時における分岐では、title#0Indexから識別子が取り出されて、その識別子の動的シナリオが再生装置により実行される。この識別子が、トップメニュータイトルの識別子なら、アプリケーション強制終了時には、自動的にトップメニューOBJが選択されることになる。

以上は本実施形態における記録媒体に対する改良である。続いて本実施形態における再生装置に対する改良について説明する。上述した記録媒体の改良に対応するため、再生装置内のモジュールマネージャ３４は図３１に示すような処理手順で処理を行う。図３１は、モジュールマネージャ３４の処理手順を示すフローチャートである。本フローチャートは、ステップＳ３１、ステップＳ３２からなるループ処理を構成しており、ステップＳ３１又はステップＳ３２のどちらかがYesになった際、対応する処理を実行するものである。

【０１１０】

ステップＳ３１は、タイトルジャンプAPIの呼び出しがあったか否かの判定である。もしタイトルジャンプAPIの呼び出しがあれば、分岐先ラベルであるタイトル番号jを取得し(ステップＳ３３)、Index Tableにおけるタイトル番号jのIndexから、IDjを取り出して(ステップＳ３４)、IDjのMovieオブジェクト又はBD-Jオブジェクトを、HDMVモジュール３３又はBD-Jモジュール３５に実行させる(ステップＳ３５)。

【０１１１】

ステップＳ３２は、タイトル終了がアプリケーションマネージャ３６から通知されたか否かの判定であり、もし通知されれば(ステップＳ３２でYes)、トップメニュータイトルを構成するトップメニューOBJをHDMVモジュール３３又はモジュールマネージャ３４に実行させる(ステップＳ３６)。

以上のアプリケーションマネージャ３６によるアプリケーション強制終了の動作例を、図３２を参照しながら説明する。ここで再生すべきタイトルは、落下するタイル片を積み重ねるというゲームアプリを含む非AV系タイトルである。図３２の下段は、アプリケーションの生存区間からなるタイトル時間軸を示し、上段は、タイトル時間軸において表示される画像を示す。非AV系タイトルがゲームアプリである場合、このゲームアプリの生存区間において、図３２の上段左側のように、ゲームアプリの一画面が表示される。ゲームアプリにバグがあり、異常終了すると、アプリケーションマネージャ３６は図２３のフローチャートに従ってゲームアプリを強制終了させ、タイトルの終了をモジュールマネージャ３４に通知する。タイトル終了が通知されると、モジュールマネージャ３４はトップメニュータイトルに分岐する。そうすると、図３２の上段右側に示すような画像が表示され、ユーザの操作待ちになる。

【０１１２】

以上のように本実施形態によれば、プログラムが含むが、デジタルストリームは含まないような非AV系タイトルの終了時においても、トップメニュータイトルに分岐するという

制御が可能になる。これによりアプリケーションプログラムがエラー終了したとしても、ブラックアウトやハングアップの発生を回避することができる。

【０１１３】

（第５実施形態）

BD-Jモードにおいて、PL再生との同期をどのように実現するかという改良に関する。図８（ｂ）の一例においてJMFプレーヤインスタンスの再生を命じるJMFプレーヤインスタンス(A.play;)をJava仮想マシン３８が解読した場合、Java仮想マシン３８はPL再生APIをコールして、コール直後に”サクセス”を示す応答をアプリケーションに返す。

【０１１４】

Playback Control Engine３２は、PL再生APIがコールされれば、PL情報に基づく処理手順を実行する。PLが2時間という再生時間を有するなら、この2時間の間、上述した処理は継続することになる。ここで問題になるのは、Java仮想マシン３８がサクセス応答を返す時間と、Playback Control Engine３２が実際に処理を終える時間とのギャップである。Java仮想マシン３８は、イベントドリブンの処理主体であるためコール直後に再生成功か、再生失敗かを示す応答を返すが、Playback Control Engine３２による実際の処理終了は2時間経過後であるので、サクセス応答をアプリケーションに返す時間を基準にしたのでは、2時間経過後にあたる処理終結を感知しえない。PL再生において早送り、巻戻し、Skipが行われると、この2時間という再生期間は2時間前後に変動することになり、処理終結の感知は更に困難になる。

【０１１５】

Playback Control Engine３２は、アプリケーションとスタンドアローンで動作するため、第３実施形態のような終了判定では、PL再生の終了をタイトル終了と解釈することができない。そこで本実施形態では、アプリケーションが終了してようがいまいが、ワークメモリ３７にJMFプレーヤインスタンスがある限り、つまり、Presentation Engine３１の制御権をBD-Jモジュール３５が掌握している間、Playback Control Engine３２から再生終結イベントを待つ。そして再生終結イベントがあれば、タイトルが終了したと解釈して、次のタイトルへの分岐を行うようモジュールマネージャ３４に通知する。こうすることにより、Playback Control Engine３２がPL再生を終結した時点を、タイトルの終端とすることができる。

【０１１６】

以降図３３～図３７のフローチャートを参照して、Playback Control Engine３２による具体的な制御手順を説明する。

図３３は、Playback Control Engine３２によるPL再生手順を示すフローチャートである。この再生手順は、Presentation Engine３１に対する制御(ステップＳ４６)と、BD-ROMドライブ１又はHDD１７に対する制御(ステップＳ４８)とを主に含む。本フローチャートにおいて処理対象たるPlayItemをPlayItem#xとする。本フローチャートは、カレントPL情報(.mpls)の読み込みを行い(ステップＳ４１)、その後、ステップＳ４２～ステップＳ５０の処理を実行するというものである。ここでステップＳ４２～ステップＳ５０は、ステップＳ４９がYesになるまで、カレントPL情報を構成するそれぞれのPI情報について、ステップＳ４３～ステップＳ５０の処理を繰り返すというループ処理を構成している。このループ処理において処理対象となるPlayItemを、PlayItem#x(PI#x)とよぶ。このPlayItem#xは、カレントPLの先頭のPlayItemに設定されることにより、初期化される(ステップＳ４２)。上述したループ処理の終了要件は、このPlayItem#xがカレントPLの最後のPlayItemになることであり(ステップＳ４９)、もし最後のPlayItemでなければ、カレントPLにおける次のPlayItemがPlayItem#xに設定される(ステップＳ５０)。

【０１１７】

ループ処理において繰り返し実行されるステップＳ４３～ステップＳ５０は、PlayItem#xのClip_information_file_nameで指定されるClip情報をシナリオメモリ２１に読み込み(ステップＳ４３)、PlayItem#xのIn_timeを、カレントClip情報のEPmapを用いて、Iピクチャアドレスuに変換し(ステップＳ４４)、PlayItem#xのOut_timeを、カレントClip情報

のEP_mapを用いて、Iピクチャアドレスvに変換して(ステップＳ４５)、これらの変換で得られたアドレスvの次のIピクチャを求めて、そのアドレスの1つ手前をアドレスwに設定し(ステップＳ４７)、そうして算出されたアドレスwを用いて、IピクチャアドレスuからアドレスwまでのTSパケットの読み出しをBD-ROMドライブ１又はHDD１７に命じるというものである(ステップＳ４８)。

【０１１８】

一方、Presentation Engine３１に対しては、カレントPLMarkのmark_time_stampからPlayItem#xのOut_timeまでの出力を命じる(ステップＳ４６)。以上のステップＳ４５～ステップＳ４８により、AVClipにおいて、PlayItem#xにより指示されている部分の再生がなされることになる

その後、PlayItem#xがカレントPLの最後のPIであるかの判定がなされる(ステップＳ４９)。

【０１１９】

PlayItem#xがカレントPLの最後のPIでなければ、カレントPLにおける次のPlayItemを、PlayItem#xに設定して(ステップＳ５０)、ステップＳ４３に戻る。以上のステップＳ４３～ステップＳ５０を繰り返すことにより、PLを構成するPIは順次再生されることになる。

図３４は、アングル切り換え手順及びSkipBack,SkipNextの手順を示すフローチャートである。本フローチャートは、図３３の処理手順と並行してなされるものであり、ステップＳ５１～Ｓ５２からなるループ処理を繰り返すというものである。本ループにおけるステップＳ５１は、アングル切り換えを要求するAPIが、Java仮想マシン３８からコールされたか否かの判定であり、アングル切り換えAPIのコールがあれば、カレントClip情報を切り換えるという操作を実行する。

【０１２０】

図３４のステップＳ５５は、判定ステップであり、PlayItem#xのis_multi_anglesがオンであるか否かの判定を行う。is_multi_anglesとは、PlayItem#xがマルチアングルに対応しているか否かを示すフラグであり、もしステップＳ５５がNoであるならステップＳ５３に移行する。ステップＳ５５がYesであるなら、ステップＳ５６～ステップＳ５９を実行する。ステップＳ５６～ステップＳ５９は、切り換え後のアングル番号を変数yに代入して(ステップＳ５６)、PlayItem#xにおけるy番目のClip_information_file_nameで指定されているClip情報をシナリオメモリ２１に読み出し(ステップＳ５７)、カレントPTMを、カレントClip情報のEP_mapを用いてIピクチャアドレスuに変換し(ステップＳ５８)、PlayItem#xのOut_timeを、カレントClip情報のEP_mapを用いてIピクチャアドレスvに変換する(ステップＳ５９)というものである。こうしてIピクチャアドレスu,vを変化した後、ステップＳ４６に移行する。ステップＳ４６への移行により、別のAVClipからTSパケットが読み出されるので、映像内容が切り換わることになる。

【０１２１】

一方、図３４のループにおけるステップＳ５２は、SkipBack/SkipNextを意味するAPIがJava仮想マシン３８からコールされたか否かの判定であり、もしコールされれば、図３５のフローチャートの処理手順を実行する。図３５は、SkipBack,SkipNextAPIがコールされた際の処理手順を示すフローチャートである。SkipBack,SkipNextを実行するにあたっての処理手順は多種多様なものである。ここで説明するのはあくまでも一例に過ぎないことに留意されたい。

【０１２２】

ステップＳ６１は、PSRで示されるカレントPI番号、及び、カレントPTMを変換することにより、カレントMark情報を得る。ステップＳ６２は、押下されたのがSkipNextキーであるか、SkipBackキーであるかの判定であり、SkipNextキーであるならステップＳ６３において方向フラグを+1に設定し、SkipBackキーであるならステップＳ６４において方向フラグを-1に設定する。

【０１２３】

ステップＳ６５は、カレントPLMarkの番号に方向フラグの値を足した番号を、カレントPLMarkの番号として設定する。ここでSkipNextキーであるなら方向フラグは+1に設定されているのでカレントPLMarkはインクリメントされることになる。SkipBackキーであるなら方向フラグは-1に設定されているので、カレントPLMarkはデクリメントされることになる。

【０１２４】

ステップＳ６６では、カレントPLMarkのref_to_PlayItem_Idに記述されているPIを、PlayItem#xに設定し、ステップＳ６７では、PlayItem#xのClip_information_file_nameで指定されるClip情報を読み込む。ステップＳ６８では、カレントClip情報のEP_mapを用いて、カレントPLMarkのmark_time_stampを、Iピクチャアドレスuに変換する。一方ステップＳ６９では、PlayItem#xのOut_timeを,カレントClip情報のEP_mapを用いて,Iピクチャアドレスvに変換する。ステップＳ７０は、カレントPLMarkのmark_time_stampからPlayItem#xのOut_timeまでの出力をPresentation Engine３１に命じた上で、図３３のステップＳ４７に移行する。こうしてIピクチャアドレスu,vを変化して、別の部分の再生を命じた上でステップＳ４７への移行するので、別のAVClipからTSパケットが読み出されることになり、映像内容が切り換えが実現する。

【０１２５】

図３６は、Presentation Engine３１による処理手順の詳細を示すフローチャートである。本フローチャートは、IピクチャのPTSをカレントPTMに設定した後で(ステップＳ７１)、ステップＳ７２～ステップＳ７７からなるループ処理を実行するものである。

続いてステップＳ７２～ステップＳ７７におけるループ処理について説明する。このループ処理は、カレントPTMにあたるピクチャ、オーディオの再生出力と、カレントPTMの更新とを繰り返すものである。本ループ処理におけるステップＳ７６は、ループ処理の終了要件を規定している。つまりステップＳ７６は、カレントPTMがPI#xのOut_timeであることをループ処理の終了要件にしている。

【０１２６】

ステップＳ７３は、早送りAPI、又は、早戻しAPIがJava仮想マシン３８からコールされたか否かの判定である。もしコールされれば、ステップＳ７８において早送りか早戻しかの判定を行い、早送りであるなら、次のIピクチャのPTSをカレントPTMに設定する(ステップＳ７９)。このようにカレントPTMを、次のIピクチャのPTSに設定することで、1秒飛びにAVClipを再生してゆくことができる。これにより、2倍速等でAVClipは順方向に早く再生されることになる。早戻しであるなら、カレントPTMがPlayItem#xのOut_timeに到達したかを判定する(ステップＳ８０)。もし到達してないのなら、1つ前のIピクチャのPTSをカレントPTMに設定する(ステップＳ８１)。このように読出先アドレスAを、1つ前のIピクチャに設定することで、AVClipを後方向に1秒飛びに再生してゆくことができる。これにより、2倍速等でAVClipは、逆方向に再生されることになる。尚、早送り、巻戻しを実行するにあたっての処理手順は多種多様なものである。ここで説明するのはあくまでも一例に過ぎないことに留意されたい。

【０１２７】

ステップＳ７４は、メニューコールAPIがコールされたか否かの判定であり、もしコールされれば、現在の再生処理をサスペンドして(ステップＳ８２)、メニュー処理用のメニュープログラムを実行する(ステップＳ８３)。以上の処理により、メニューメニューコールがなされた場合は、再生処理を中断した上で、メニュー表示のための処理が実行されることになる。

【０１２８】

ステップＳ７５は、sync_PlayItem_idにより、PlayItem#xを指定したSubPlayItem#yが存在するか否かの判定であり、もし存在すれば、図３７のフローチャートに移行する。図３７は、SubPlayItemの再生手順を示すフローチャートである。本フローチャートでは、先ずステップＳ８６において、カレントPTMはSubPlayItem#yのsync_start_PTS_of_playItemであるか否かを判定する。もしそうであれば、ステップＳ９３においてSubPlayItem#y

に基づく再生処理を行うようPlayback Control Engine３２に通知する。

【０１２９】

図３７のステップＳ８７～ステップＳ９２は、SubPlayItem#yに基づく再生処理を示すフローチャートである。

ステップＳ８７では、SubPlayItem#yのClip_information_file_nameで指定されるClip情報を読み込む。ステップＳ８８では、カレントClip情報のEP_mapを用いて、SubPlayItem#yのIn_timeを、アドレスαに変換する。一方ステップＳ８９では、SubPlayItem#yのOut_timeを,カレントClip情報のEP_mapを用いて、アドレスβに変換する。ステップＳ９０は、SubPlayItem#yのIn_timeからSubPlayItem#yのOut_timeまでの出力をデコーダに命じる。これらの変換で得られたアドレスβの次のIピクチャを求めて、そのアドレスの1つ手前をアドレスγに設定し(ステップＳ９１)、そうして算出されたアドレスγを用いて、SubClip#zにおけるアドレスαからアドレスγまでのTSパケットの読み出しをBD-ROMドライブ１又はHDD１７に命じるというものである(ステップＳ９２)。

【０１３０】

また図３３に戻ってPlayback Control Engine３２の処理の説明の続きを行う。ステップＳ５３はPresentation Engine３１による再生制御が完了したかの判定であり、最後のPlayItem#xに対して、図３６のフローチャートの処理が行われている限り、ステップＳ５３がNoになる。図３６のフローチャートの処理が終了して初めて、ステップＳ５３はYesになりステップＳ５４に移行する。ステップＳ５４は、Java仮想マシン３８への再生終結イベントの出力であり、この出力により、2時間という再生時間の経過をJava仮想マシン３８は知ることができる。

【０１３１】

以上が本実施形態におけるPlayback Control Engine３２、Presentation Engine３１の処理である。続いて本実施形態におけるアプリケーションマネージャ３６処理手順について説明する。図３８は、第５実施形態に係るアプリケーションマネージャ３６の処理手順を示すフローチャートである。

図３８のフローチャートは、図２７のフローチャートを改良したものである。その改良点は、ステップＳ２１－ステップＳ２２間にステップＳ２４が追加され、このステップＳ２４がYesになった際、実行されるステップＳ１０１が存在する点である。

【０１３２】

ステップＳ２４は、JMFプレーヤインスタンスがワークメモリ３７に存在するか否かの判定であり、もし存在しなければステップＳ２２に移行する。存在すれば、ステップＳ１０１に移行する。ステップＳ１０１は、Playback Control Engine３２から再生終結イベントが出力されたか否かの判定であり、もし出力されれば、ワークメモリ中のJavaプレーヤインスタンスを消滅させた上で(ステップＳ１０２)、タイトル終了をモジュールマネージャ３４に通知する(ステップＳ２６)。通知されねば、ステップＳ２１～ステップＳ２４からなるループ処理を繰り返す。

【０１３３】

以上のフローチャートにおいて、ワークメモリ３７にJMFプレーヤインスタンスが存在する限り(ステップＳ２４でYes)、ステップＳ２２、ステップＳ２３はスキップされる。そのため、たとえ全てのアプリケーションが終了したとしてもタイトルは継続中と解釈される。

以上のように本実施形態によれば、2時間という再生時間の経過時点をアプリケーションマネージャ３６は把握することができるので、PL再生の終了条件にメニューを表示して、このメニューに対する操作に応じて他のタイトルに分岐するという制御を実現することができる。

【０１３４】

(第６実施形態)

第６実施形態は、BD-Jオブジェクトにデータ管理テーブルを設ける改良に関する。

データ管理テーブル(DMT)は、そのタイトル時間軸においてローカルメモリ２９上にロ

ードすべきJavaアーカイブファイルを、読込属性と、読込優先度とに対応づけて示すテーブルである。”ローカルメモリ２９における生存”とは、そのアプリケーションを構成するJavaアーカイブファイルがローカルメモリ２９から読み出され、Java仮想マシン３８内のワークメモリ３７への転送が可能になっている状態をいう。図３９は、データ管理テーブルの一例を示す図である。本図に示すようにデータ管理テーブルは、アプリケーションの『生存区間』と、その生存区間をもったアプリケーションを識別する『applicationID』と、そのアプリケーションの『読込属性』と、『読込優先度』とを示す。

【０１３５】

上述したようにアプリケーション管理テーブルには、生存区間という概念があり、データ管理テーブルにも同じ生存区間という概念がある。アプリケーション管理テーブルと同じ概念を、データ管理テーブルに設けておくというのは一見無駄のように思えるがこれには意図がある。

図４０は、BD-Jオブジェクトが想定している実行モデルを示す図である。本図における実行モデルは、BD-ROM、ローカルメモリ２９、Java仮想マシン３８からなり、BD-ROM、ローカルメモリ２９、ワークメモリ３７という三者の関係を示す。矢印my1は、BD-ROM→ローカルメモリ２９間の読み込みを示し、矢印my2は、ローカルメモリ２９→ワークメモリ３７間の読み込みを示す。矢印上の注釈は、これらの読み込みが、どのようなタイミングでなされるかを示す。注釈によると、BD-ROM→ローカルメモリ２９間の読み込みは、いわゆる”先読み”であり、アプリケーションが必要となる以前の時点に行われねばならない。

【０１３６】

また注釈によると、ローカルメモリ２９→ワークメモリ３７間の読み込みは、アプリケーションが必要になった際になされることがわかる。”必要になった際”とは、アプリケーションの生存区間が到来した時点(1)、アプリケーションの呼出が他のアプリケーション又はアプリケーションマネージャ３６から指示された時点(2)を意味する。

矢印my3は、ワークメモリ３７におけるアプリケーションの占有領域の解放を示し、矢印my4は、ローカルメモリ２９におけるアプリケーションの占有領域の解放を示す。矢印上の注釈は、これらの読み込みが、どのようなタイミングでなされるかを示す。注釈によると、ワークメモリ３７上の解放は、アプリケーション終了と同時になされることがわかる。一方ローカルメモリ２９上の解放は、Java仮想マシン３８にとって必要でなくなった時点でなされる。この必要でなくなった時点とは、”終了時点”ではない。”終了した上、再起動の可能性もない時点”であること、つまり該当するtitleが終了した時点を意味する。上述した読込・解放のうち、ワークメモリ３７における解放時点は、アプリケーション管理テーブルにおける生存区間から判明する。しかし”アプリケーションが必要となる以前の時点”、”終了した上、再起動の可能性もない時点”については、規定し得ない。そこで、オーサリング段階において、かかる時点をディスクコンテンツ全体の時間軸上で規定しておくため、本実施形態では各アプリケーションが生存している区間を、アプリケーション管理テーブルとは別に、データ管理テーブルに記述するようにしている。つまり”アプリケーションが必要となる以前の時点”をデータ管理テーブルにおける生存区間の始点と定義し、”終了した上、再起動の可能性もない時点”をデータ管理テーブルの終点と定義することにより、上述したローカルメモリ２９上の格納内容の遷移をオーサリング時に規定しておくことができる。これがデータ管理テーブルの記述意義である。

【０１３７】

データ管理テーブルによるローカルメモリ２９生存区間の記述について説明する。ここで制作しようとするディスクコンテンツは3つのタイトル(title#1、title#2、title#3)からなり、これらタイトルの時間軸において、図４１（ｂ）に示すようなタイミングで、ローカルメモリ２９を使用したいと考える。この場合、title#1時間軸の開始点においてapplication#1、application#2を構成するJavaアーカイブファイルをローカルメモリ２９に読み込み、title#1時間軸の継続中、application#1、application#2をローカルメモリ２９に常駐させておく。そしてtitle#2時間軸の始点で、application#1を構成するJavaアー

カイブファイルをローカルメモリ２９から解放して、代わりにapplication#3を構成するJavaアーカイブファイルをローカルメモリ２９に読み込んで、常駐させるというものである(以降、アプリケーションを構成するJavaアーカイブファイルは、アプリケーションと同義に扱う。)。この場合のデータ管理テーブルの記述は、図４１（ａ）の通りであり、アプリケーションのapplicationIDを、その生存区間に対応づけて記述することで、ローカルメモリ２９に常駐すべきアプリケーションを表現する。図４１（ａ）では、application#1のapplicationIDがtitle#1と対応づけられて記述されており、application#2のapplicationIDはtitle#1、title#2と対応づけられ、application#3のapplicationIDはtitle#3と対応づけられて記述されていることがわかる。こうすることで、ローカルメモリ２９占有の時間的遷移がオーサリング担当者により規定されることになる。

【０１３８】

データ管理テーブル、アプリケーション管理テーブルの組合せとしては、アプリケーション管理テーブルに規定する生存区間は、細かい再生単位にし、データ管理テーブルに規定する生存区間は、大まかな再生単位にすることが望ましい。大まかな再生単位には、タイトル、PLといった非シームレスな再生単位が望ましい。一方、細かい再生単位としては、PL内のチャプターというようにシームレスな再生単位が望ましい。アプリケーションの生存区間をタイトル毎、PL毎に定めれば、アプリケーションはローカルメモリ２９上に存在するので、そのタイトルの再生中においてアプリケーションは何時でも取り出せる状態になる。そうであれば、アプリケーションの生存区間を細かく定めたとしても、アプリケーションを即座に、仮想マシン上のワークメモリに読み出すことができるので、アプリケーションの起動・終了が頻繁になされたとしても、スムーズなアプリケーション実行を実現することができる。

【０１３９】

次に、読込属性について説明する。

図２においてJavaアーカイブファイルは、AVClipとは別の記録領域に記録されることを前提にしていた。しかしこれは一例に過ぎない。Javaアーカイブファイルは、BD-ROMにおいてAVClipが占める記録領域に埋め込まれることがある。この埋め込みの態様には、カルーセル化、インターリーブユニット化という2種類がある。

【０１４０】

ここで”カルーセル化”とは、対話的な放送の実現のために同一内容を繰り返しするという放送方式に変換することである。BD-ROMは、放送されたデータを格納するものではないが、本実施形態では、カルーセルの放送形式に倣ってJAVAアーカイブファイルを格納するようにしている。図４２は、カルーセル化によるJavaアーカイブファイル埋め込みを示す図である。第1段目は、AVClip中に埋め込むJavaアーカイブファイルであり、第2段目は、セクション化を示す。第3段目は、TSパケット化、第4段目は、AVClipを構成するTSパケット列を示す。こうしてセクション化、TSパケット化されたデータ(図中の”D”)が、AVClipに埋め込まれるのである。カルーセルによりAVClipに多重化されたJavaアーカイブファイルは、読み出すにあたって、低帯域で読み出されることになる。この低帯域での読み出しは、概して2～3分というように長期間を要するため、再生装置はJavaアーカイブファイルを2～3分をかけて読み込むことになる。

【０１４１】

図４３は、インターリーブ化によるJavaアーカイブファイル埋め込みを示す図である。第1段目は、埋め込まれるべきAVClip、第2段目は、AVClipにインターリーブ化されたJavaアーカイブファイル、第3段目は、BD-ROMの記録領域におけるAVClip配置である。本図に示すように、ストリームに埋め込まれるべきJavaアーカイブファイルは、インターリーブ化され、AVClipを構成するXXXXX.m2tsを構成する分割部分(図中のAVClip2/4,3/4)の合間に記録される。インターリーブ化によりAVClipに多重化されたJavaアーカイブファイルは、カルーセル化と比較して、高い帯域で読み出されることになる。この高い帯域での読み出しであるため、再生装置はJavaアーカイブファイルを比較的短期間に読み込むことになる。

【０１４２】
カルーセル化・インターリーブ化されたJavaアーカイブファイルは、プリロードされるのではない。BD-ROMにおけるAVClipの記録領域のうち、カルーセル化・インターリーブ化されたJavaアーカイブファイルが埋め込まれた部分に、現在の再生時点が到達した際、再生装置のローカルメモリ２９にロードされる。Javaアーカイブファイルの記録態様には、図２に示すものの他に、図４２、図４３（ａ）に示すものがあるので、読込属性は、図４３（ｂ）に示すように、設定されうる。図４３（ｂ）に示すように、読込属性は、タイトル再生に先立ち、ローカルメモリ２９に読み込まれる旨を示す”Preload”と、タイトル再生中に、カルーセル化方式で読み込まれる旨を示す”Load.Carousel”と、タイトル再生中に、インターリーブ化方式で読み込まれる旨を示す”Load.InterLeave”とがある。読込属性には、カルーセル化されているか、インターリーブ化されているかが添え字で表現されているが、これを省略してもよい。

【０１４３】
データ管理テーブルにおける生存区間の具体的な記述例について、図４４を参照しながら説明する。図４４（ａ）は、データ管理テーブルの一例を示す図である。図４４（ｂ）は、かかるデータ管理テーブルの割り当てによるローカルメモリ２９の格納内容の変遷を示す図である。本図は、縦軸方向にローカルメモリ２９における占有領域を示し、横軸を、1つのタイトル内のPL時間軸としている。データ管理テーブルにおいてapplication#1は、1つのタイトル内のPL時間軸全体を生存区間とするよう記述されているので、このタイトルのChapter#1～Chapter#5においてローカルメモリ２９内の領域を占有することになる。データ管理テーブルにおいてapplication#2は、タイトル内のPL#1におけるChapter#1～Chapter#2を生存区間とするよう記述されているので、このタイトルのChapter#1～Chapter#2においてローカルメモリ２９内の領域を占有することになる。データ管理テーブルにおいてapplication#3は、タイトル内のPL#1におけるChapter#4～Chapter#5を生存区間とするよう記述されているので、このタイトルのChapter#4～Chapter#5においてローカルメモリ２９内の領域を占有することになる。以上で、データ管理テーブルにおける生存区間についての説明を終える。

続いて読込優先度について説明する。読込優先度とは、ローカルメモリ２９への読み込みに対する優劣を決める優先度である。読込優先度には複数の値がある。2段階の優劣を設けたい場合、Mandatoryを示す値、optionalを示す値を読込優先度に設定する。この場合、Mandatoryは高い読込優先度を意味し、optionalは、低い読込優先度を意味する。3段階の優劣を設けたい場合、Mandatoryを示す値、optional:high、optional:lowを示す値を読込優先度に設定する。Mandatoryは、最も高い読込優先度を示し、optional:highは、中程度の読込優先度、optional:lowは、最も低い読込優先度を示す。データ管理テーブルにおける読込優先度の具体的な記述例について、図４５（ａ）（ｂ）を参照しながら説明する。この具体例で、想定しているローカルメモリ２９のメモリ規模は、図４５（ａ）に示すようなものである。図４５（ａ）は、新旧再生装置におけるローカルメモリ２９のメモリ規模を対比して示す図である。矢印mk1は旧再生装置におけるメモリ規模を、矢印mk2は新再生装置におけるメモリ規模をそれぞれ示す。この矢印の対比から、新再生装置におけるローカルメモリ２９のメモリ規模は、旧再生装置のそれと比較して、三倍以上である状態を想定している。このようにメモリ規模にバラツキがある場合、アプリケーションは、図４５に示すような2つのグループに分類される。1つ目は、どのようなメモリ規模であっても読み込むんでおくべきアプリケーション(#1,#2)である。2つ目は、旧再生装置での読み込みは望まないが、新再生装置での読み込みは希望するアプリケーション(#3,#4)である。読み込もうとするアプリケーションが、これら2つのグループに分類されれば、前者に帰属するアプリケーションに、読込優先度=Mandatoryを設定し、後者に属するアプリケーションに、読込優先度=Optionalを設定する。図４５（ｂ）は、読込優先度が設定されたデータ管理テーブルの一例を示す図である。データ管理テーブルをこのように設定した上で、application#1～application#4をBD-ROMに記録すれば、あらゆるメモリ規模の再生装置での再生を保証しつつも、メモリ規模が大きい再生装置では、より大きなサイズのデ

ータを利用したアプリケーションを再生装置に再生させることができる。

【０１４４】

以上は本実施形態における記録媒体に対する改良である。続いて本実施形態における再生装置に対する改良について説明する。上述した記録媒体の改良に対応するため、アプリケーションマネージャ３６は図４６に示すような処理手順で処理を行う。

図４６は、アプリケーションマネージャ３６によるプリロード制御の処理手順を示す図である。本フローチャートは、再生すべきタイトルにおけるデータ管理テーブルを読み込み(ステップＳ１１１)、データ管理テーブルにおいて最も高い読込優先度をもちつつ、applicationIDが最も小さいアプリケーションをアプリケーションiにした上で(ステップＳ１１２)、ステップＳ１１３、ステップＳ１１４の判定を経た上で、アプリケーションiをローカルメモリ２９にプリロードする(ステップＳ１１５)という処理を、ステップＳ１１６がNo及びステップＳ１１７がNoと判定されるまで、繰り返すというループ処理を構成している。

【０１４５】

ステップＳ１１３は、アプリケーションiの読込属性がプリロードであるか否かの判定であり、ステップＳ１１４は、アプリケーションの読込優先度が=Mandatoryであるか0ptionalであるかの判定である。ステップＳ１１３においてプリロードと判定され、ステップＳ１１４において読込優先度がMandatoryと判定されれば、アプリケーションはローカルメモリ２９にプリロードされることになる(ステップＳ１１５)。もしステップＳ１１３において読込属性がロードであると判定されれば、ステップＳ１１４～ステップＳ１１５はスキップされることになる。

【０１４６】

ループ処理の終了要件を規定する2つのステップのうちステップＳ１１６は、applicationIDが次に高く、アプリケーションiと同一読込優先度のアプリケーションkが存在するか否かを判定するものである。そのようなアプリケーションkが存在するなら、そのアプリケーションkをアプリケーションiにする(ステップＳ１１９)。

ループ処理の終了要件を規定する2つのステップのうちステップＳ１１７は、データ管理テーブルにおいて次に低い読込優先度をもつアプリケーションが存在するか否かの判定であり、もし存在すれば、その次に低い読込優先度をもつアプリケーションのうち、最も小さいapplicationIDをアプリケーションkを選んで(ステップＳ１１８)、そのアプリケーションkをアプリケーションiにする(ステップＳ１１９)。これらステップＳ１１６、ステップＳ１１７がYesになっている限り、上述したステップＳ１１３～ステップＳ１１５の処理は繰り返されることになる。ステップＳ１１６、ステップＳ１１７において、該当するアプリケーションが無くなれば本フローチャートの処理は終了することになる。

【０１４７】

ステップＳ１２０～ステップＳ１２３は、ステップＳ１１４において読込優先度＝0Optionalであると判定された場合に、実行される処理である。

ステップＳ１２０は、同じapplicationIDをもち、読込優先度が高いアプリケーションjが存在するか否かの判定である。

ステップＳ１２１は、ローカルメモリ２９の残り容量がアプリケーションiのサイズを上回るか否かを判定するステップである。ステップＳ１２０がNo、ステップＳ１２１がYesである場合、ステップＳ１１５においてアプリケーションiがローカルメモリ２９にプリロードされることになる。ステップＳ１２０がNo、ステップＳ１２１がNoである場合、アプリケーションiはローカルメモリ２９にプリロードされずそのままステップＳ１１６に移行することになる。

【０１４８】

こうしておくと、読込優先度＝0Optionalのデータは、ステップＳ１２０－ステップＳ１２１の判定がYesにならないと、ローカルメモリ２９へのプリロードがなされない。メモリ規模が小さい旧再生装置は、2～3個のアプリケーションを読み込んだ程度で、ステップＳ１２１の判定はNoになるが、メモリ規模が大きい新再生装置は、更に多くのアプリケー

ションを読み込んだとしても、ステップＳ１２１の判定はNoにならない。以上のように、旧再生装置では、ローカルメモリ２９にMandatoryのアプリケーションのみが読み込まれ、新再生装置には、Mandatoryのアプリケーションと、Optionalのアプリケーションとが読み込まれることになる。

【０１４９】

ステップＳ１２２は、ステップＳ１２０においてYesと判定された場合に実行されるステップである。同じapplicationIDをもち、読込優先度が高いアプリケーションjがローカルメモリ２９上に存在する場合、ローカルメモリ２９の残り容量と、アプリケーションjのサイズとの和が、アプリケーションiのサイズを上回るか否かを判定し(ステップＳ１２２)、もし上回れば、アプリケーションiを用いてローカルメモリ２９上のアプリケーションjを上書きすることによりプリロードする(ステップＳ１２３)。下回る場合は、アプリケーションiはローカルメモリ２９にプリロードされずそのままステップＳ１１６に移行することになる。

【０１５０】

ステップＳ１１５、ステップＳ１２３による読込処理の一例を、図４７（ａ）を参照しながら説明する。図４７（ａ）は、この具体例が想定しているデータ管理テーブルの一例を示す図である。本図における3つのアプリケーションは、それぞれ3つのファイルに格納されており、applicationIDは同じであるが(applicationID=1)、読込優先度は互いに異なる(mandatory,optional:high,optional:low)。こうしたデータ管理テーブルが処理対象であると、ステップＳ１１５により、読込優先度＝Mandatoryのアプリケーションはローカルメモリ２９に読み込まれる。しかし読込優先度＝Optionalのアプリケーションについては、ステップＳ１２０～ステップＳ１２２の判定を経た上で、ステップＳ１２３において読み込まれる。ステップＳ１１５と違いステップＳ１２３では、既にローカルメモリ２９にある同じapplicationIDのアプリケーションを上書きしてゆくよう、プリロードがなされるので、複数アプリケーションのうち1つが排他的に、ローカルメモリ２９にロードされることになる。

i)読込優先度＝mandatoryのアプリケーションを読み込んだ後、読込優先度＝optional:highのアプリケーションを読み込むにあたって、ステップＳ１２２がNoと判定されればれば、読込優先度＝mandatoryのアプリケーションがローカルメモリ２９に残ることになる。読込優先度＝mandatoryのアプリケーションを読み込んだ後、読込優先度＝optional:highのアプリケーションを読み込むにあたって、ステップＳ１２２がYesと判定されれば、読込優先度＝optional:highのアプリケーションにより、読込優先度＝mandatoryのアプリケーションは上書きされ、読込優先度＝optional:highのアプリケーションがローカルメモリ２９に残ることになる。

【０１５１】

ii)読込優先度＝optional:highのアプリケーションを読み込んだ後、読込優先度＝optional:lowのアプリケーションを読み込むにあたって、ステップＳ１２２がNoと判定されればれば、読込優先度＝Mandatoryのアプリケーションがローカルメモリ２９に残ることになる。読込優先度＝optional:highのアプリケーションを読み込んだ後、読込優先度＝optional:lowのアプリケーションを読み込むにあたって、ステップＳ１２２がYesと判定されれば、読込優先度＝optional:lowのアプリケーションにより、読込優先度＝optional:highのアプリケーションは上書きされ(ステップＳ１２３)、読込優先度＝optional:lowのアプリケーションがローカルメモリ２９に残ることになる。

【０１５２】

ローカルメモリ２９の容量が許す限り、ローカルメモリ２９上のアプリケーションを上書きしてゆくとの処理が繰り返されるので、ローカルメモリ２９の格納内容は、図４７（ｂ）に示すように、mandatory＝optional:high＝>optional:lowと遷移してゆくことになる。メモリ規模に応じて、サイズが異なるJavaアーカイブファイルをローカルメモリ２９にロードすることができるので、メモリ規模が小さい再生装置については、必要最小限の解像度をもったサムネール画像を有するJavaアーカイブファイルを、メモリ規模が中程度

の再生装置については、中程度の解像度をもったSD画像を有するJavaアーカイブファイルを、メモリ規模が大規模である再生装置については、高解像度をもったHD画像を有するJavaアーカイブファイルをローカルメモリ２９にロードすることができる。かかるロードにより、メモリ規模に応じて解像度が異なる画像を表示させることができ、オーサリング担当者によるタイトル制作の表現の幅が広がる。

図４８は、データ管理テーブルを参照した読取処理の具体例を示す図である。本図における2つのアプリケーションは、同じapplicationID(application#3)が付与された2つのアプリケーションを示す図である。そのうち一方は、AVClip中に埋め込まれていて、読込優先度がmandatoryに設定されている。他方は、AVClipとは別ファイルに記録されていて、読込優先度がOptionalに設定されている。前者のアプリケーションは、AVClipに埋め込まれているので、その埋込部分にあたる生存区間が、生存区間(title#1:chapter#4～#5)として記述されている。これらのアプリケーションのうちapplication#2、application#3には、ロードを示す読込属性が付与されている。application#2はChapter#1～Chapter#2を生存区間にしており、application#3はChapter#4～Chapter#5を生存区間にしているので、タイトル時間軸においてどちらか一方が排他的にローカルメモリ２９上に常駐することになる。図４８（ｂ）は、タイトル時間軸上の別々の時点において、排他的に格納されるapplication#2、application#3を示す図である。これは必要最低限のメモリ規模しかもたない再生装置での再生を念頭に置いた配慮である。こうした内容のデータ管理テーブルが処理対象であるとアプリケーションマネージャ３６は、上述した図４６のフローチャートによりメモリ規模に応じて異なる処理を行う。

【０１５３】

後者のアプリケーションは、読込優先度＝ロードであるので、ローカルメモリ２９にロードされる。かかる処理により、Mandatoryなメモリ規模さえあれば、アプリケーションマネージャはデータをローカルメモリ２９にロードすることができる。ここで問題になるのは、メモリ規模が大きい再生装置による読み込み時である。メモリ規模が大きいにも拘らず、Chapter#4～Chapter#5に到達するまでapplication#3を読み込めないというのは、メモリ規模の無駄になる。そこで本図のデータ管理テーブルには、同じapplication#3にプリロードを示す読込属性を付与してBD-ROMに記録しておき、これらに同じapplicationIDを付与している。

【０１５４】

前者のアプリケーションは、読込優先度＝Optionalであるので、ステップＳ１２１がYesになった場合に限り、プリロードされる(ステップＳ１１５)。こうすることで、メモリ規模が大きい再生装置は、title#1、Chapter#4～Chapter#5の到達を待つことなく、AVClipに埋め込まれているのと同じアプリケーションをローカルメモリ２９にロードすることができるのである(図４８（ｃ）)。

【０１５５】

以上がプリロード時における処理である。続いてロード時における処理手順について説明する。

図４９は、データ管理テーブルに基づくロード処理の処理手順を示す図である。本フローチャートは、ステップＳ１３１～ステップＳ１３３からなるループ処理を、タイトル再生が継続されている間、繰り返すというものである。

【０１５６】

ステップＳ１３１は、AutoRunを示す起動属性を有したアプリケーションの生存区間が到来したか否かの判定である。もし到来すれば、AutoRunを示す起動属性を有したアプリケーションをアプリケーションqにして(ステップＳ１３４)、アプリケーションqを起動する旨の起動指示をJava仮想マシン３８に発行して、アプリケーションqをローカルメモリ２９からワークメモリ３７に読み出させる(ステップＳ１３５)。

【０１５７】

ステップＳ１３３は、タイトル内PLの再生が全て終了したかの判定である。この判定は、第５実施形態に示したように、Playback Control Engine３２からの再生終結イベント

があったか否かでなされる。もし終了すれば、本フローチャートの処理を終了する。

ステップＳ１３２は、起動中アプリケーションからの呼出があったか否かの判定である。もしあれば、呼出先アプリケーションをアプリケーションqにして(ステップＳ１３６)、現在の再生時点は、アプリケーション管理テーブルにおけるアプリケーションqの生存区間であるか否かを判定する(ステップＳ１３７)。もし生存区間でなければ、起動失敗を表示して(ステップＳ１４８)、ステップＳ１３１～ステップＳ１３３からなるループ処理に戻る。生存区間であれば、図５０のフローチャートに従い、ロード処理を行う。

【０１５８】

図５０におけるステップＳ１３８は、現在の再生時点がデータ管理テーブルにおけるアプリケーションqの生存区間であるか否かを示す判定である。もし生存区間でなければ、アプリケーションqはローカルメモリ２９にロードすることができない。この場合、アプリケーションqを起動する旨の起動指示をJava仮想マシン３８に発行し、ローカルメモリ２９を介することなく、直接アプリケーションqをBD-ROMからワークメモリ３７に読み出させる。この場合アプリケーションを読み出すためのヘッドシークが発生するから、PL再生は中断することになる(ステップＳ１４５)。

【０１５９】

もし生存区間であれば、ステップＳ１３９において、アプリケーションには読込属性が付加されているか否かを判定する。読込属性がないということは、アプリケーションqは、カルーセル化、若しくはインターリーブ化されていないことを意味する。しかし読込属性が付加されていなくても、ローカルメモリ２９にアプリケーションqを置くことは許される。そこで再生中断を承知の上、アプリケーションの読み出しを行う。つまりBD-ROMからローカルメモリ２９へとアプリケーションを読み出した上で、アプリケーションをワークメモリ３７に読み出す(ステップＳ１４０)。

【０１６０】

ステップＳ１４１～ステップＳ１４６は、ステップＳ１３９がYesと判定された場合になされる処理である。ステップＳ１４１では、読込属性を参照することで、アプリケーションがプリロードされているか否かを判定する。プリロードされていれば、ステップＳ１３５に移行する。

ステップＳ１４２は、読込属性がロードである場合に実行される判定ステップであり、アプリケーションqがカルーセル化されているか、インターリーブ化されているかを判定する。インターリーブ化されていれば、キャッシュセンスをJava仮想マシン３８に実行させる(ステップＳ１４３)。ローカルメモリ２９にアプリケーションqが存在すれば、ステップＳ１３５に移行して、アプリケーションqをJava仮想マシン３８にロードさせる。

【０１６１】

ローカルメモリ２９にアプリケーションがなければ、トップメニュータイトルに分岐する等の例外処理を行う(ステップＳ１４４)。カルーセル化されていれば、タイマをセットし(ステップＳ１４８)、そのタイマがタイムアウトするまで(ステップＳ１４７)、キャッシュセンスをJava仮想マシン３８に実行させる(ステップＳ１４６)。もしローカルメモリ２９にアプリケーションqが出現すれば、図４９のステップＳ１３５に移行して、アプリケーションqをJava仮想マシン３８にロードさせる。タイムアウトすれば、トップメニュータイトルに分岐する等の例外処理を行う(ステップＳ１４４)。

【０１６２】

図５１は、Java仮想マシン３８によるアプリケーションの読み込みがどのようにして行われるかを模式化した図である。

矢印1,2は、アプリケーション管理テーブルに生存していて、データ管理テーブルに生存しており、カルーセル化,インターリーブ化を示す読込属性が存在するJavaアーカイブファイルの読み込みを示す。矢印1は、ステップＳ６５、６７においてなされるローカルメモリ２９センスを示す。このローカルメモリ２９センスは、カルーセル又はインターリーブ化により埋め込まれたデータが、ローカルメモリ２９に存在するかもしれないためローカルメモリ２９内をセンスするというものである。矢印2は、ステップＳ１３５に対応

する読み込みであり、アプリケーションがローカルメモリ２９に存在していた場合の、ローカルメモリ２９からワークメモリ３７へのロードを示す。×付きの矢印は、ローカルメモリ２９にデータがない場合を示す。

【０１６３】

矢印▽1,2は、アプリケーション管理テーブルに生存しているが、データ管理テーブルに生存しておらず、読込属性が存在しないJavaアーカイブファイルの読み込みを示す。

矢印▽1は、ステップＳ１４５における読み込みに対応するものであり、Java仮想マシン３８によるBD-ROMからのダイレクトリードの要求を示す。矢印▽2はその要求による、BD-ROMからワークメモリ３７へのJavaアーカイブファイル読み出しを示す。

【０１６４】

矢印☆1,2,3は、アプリケーション管理テーブルに生存していて、データ管理テーブルに生存しているが、読込属性が存在しないJavaアーカイブファイルの読み込みを示す。

矢印☆1は、ステップＳ１４０における読み込みに対応するものであり、Java仮想マシン３８によるBD-ROMからのダイレクトリードの要求を示す。矢印☆2はその要求による、ローカルメモリ２９へのJavaアーカイブファイルの読み出しを示す。矢印☆3はローカルメモリ２９からワークメモリ３７へのJavaアーカイブファイルの読み出しを示す。

【０１６５】

以上のように本実施形態によれば、ローカルメモリ２９上で同時に常駐されるアプリケーションの数が所定数以下になるように規定しておくことができるので、ローカルメモリ２９からの読み出し時におけるキャッシュミスを極力回避することができる。キャッシュミスのないアプリケーション読み出しを保証することができるので、アプリケーション呼出時にあたては、AVClipの再生を止めてまで、BD-ROMからアプリケーションを読み出すことはなくなる。AVClip再生を途切れさせないので、AVClipのシームレス再生を保証することができる。

【０１６６】

（第７実施形態）

第３実施形態では、非AV系タイトルの時間軸をアプリケーションの生存区間に基づき定めることにした。しかしアプリケーションの動作というのは不安定であり、起動の失敗や異常終了がありうる。本実施形態は、起動失敗、異常終了があった場合のFail Safe機構を提案するものである。図５２（ａ）は、第７実施形態に係るBD-Jオブジェクトの内部構成を示す図である。図７（ｂ）と比較して本図が新規なのは、プレイリスト管理テーブルが追加されている点である。

【０１６７】

図５２（ｂ）は、プレイリスト管理テーブルの一例を示す図である。本図に示すようにプレイリスト管理テーブルは、PLの指定と、そのPLの再生属性とからなる。PLの指定は、対応するタイトルのタイトル時間軸において、再生可能となるPLを示す。PLの再生属性は、指定されたPLを、タイトル再生の開始と同時に自動再生するか否かを示す(こうして自動再生されるPLをデフォルトPLという)。

【０１６８】

次にプレイリスト管理テーブルによりタイトル時間軸がどのように規定されるかを、図５３を参照しながら説明する。図５３（ａ）は、再生属性が非自動再生を示すよう設定された場合の非AV系タイトルにおけるタイトル時間軸を示す図である。この場合、デフォルトPLは再生されないから、非AV系タイトル同様、アプリケーションの生存区間からタイトル時間軸が定まる。

【０１６９】

図５３（ｂ）は、再生属性がAutoPlayに設定された非AV系タイトルのタイトル時間軸を示す図である。再生属性がAutoPlayを示すよう設定されれば、Playback Control Engine ３２は非AV系タイトルの再生開始と同時に、デフォルトPLの再生を開始する。しかしアプリケーションが正常に動作し、正常終了したとしても、このタイトル時間軸は、PL時間軸を基準にして定められる。

【０１７０】
図５３（ｃ）は、プレイリスト管理テーブルにおいて再生属性が”AutoPlay”を示すよう設定され、アプリケーションが異常終了した場合を示す。かかる異常終了により、どのアプリケーションも動作してない状態になるが、デフォルトPLの再生は継続する。この場合も、デフォルトPLのPL時間軸がタイトル時間軸になる。
図５３（ｄ）は、プレイリスト管理テーブルにおいて再生属性が”AutoPlay”を示すよう設定され、メインアプリの起動に失敗したケースを示す。この場合も、Playback Control Engine３２によるデフォルトPL再生は、アプリケーションの起動失敗とは関係なしに行われるので、デフォルトPLの時間軸がタイトル時間軸になる。
【０１７１】
以上のようにプレイリスト管理テーブルの再生属性を、”AutoPlay”に設定しておけば、Javaアプリケーションの起動に、5～10秒という時間がかかったとしても、その起動がなされている間、”とりあえず何かが写っている状態”になる。この”とりあえず何かが写っている状態”によりタイトル実行開始時のスタートアップディレイを補うことができる。
【０１７２】
以上は本実施形態における記録媒体に対する改良である。続いて本実施形態における再生装置に対する改良について説明する。
図５２（ｃ）は、分岐先タイトルのプレイリスト管理テーブルにおいて、再生属性がAutoPlayに設定されたPLが存在する場合、再生装置がどのような処理を行うかを示す図である。本図に示すように、再生属性がAutoPlayに設定されたPLが、分岐先タイトルのプレイリスト管理テーブルに存在すれば、BD-Jモジュール３５内のアプリケーションマネージャ３６は、タイトル分岐直後にこのAutoPlayPLの再生を開始するようPlayback Control Engine３２に指示する。このように再生属性がAutoPlayのPLは、タイトル分岐直後に再生開始が命じられることになる。
【０１７３】
上述した記録媒体の改良に対応するため、アプリケーションマネージャ３６は図５４に示すような処理手順で処理を行う。
図５４は、第７実施形態に係るアプリケーションマネージャ３６の処理手順を示すフローチャートである。本フローチャートは、図３８のフローチャートにおいてステップＳ２１の前にステップＳ１０３、ステップＳ１０４を追加し、ステップＳ２１と、ステップＳ２２との間にステップＳ１００を追加し、ステップＳ２３－ステップＳ２６間に、ステップＳ１０５を追加したものである。
【０１７４】
ステップＳ１０３は、対応するタイトルのプレイリスト管理テーブルの再生属性がAutoPlayであるか否かの判定である。もしAutoPlayなら、デフォルトPLに対する再生制御をPlayback Control Engine３２に開始させる(ステップＳ１０４)。
ステップＳ１００は、Presentation Engine３１による再生中であるか否かを判定する。もし再生中であるなら、ステップＳ１０１に移行する。
【０１７５】
ステップＳ１０５は、ステップＳ２３がYes、ステップＳ２５がNoである場合に実行される判定ステップであり、再生属性がAutoPlayであるか否かを示す。もし否であるなら、タイトル終了をモジュールマネージャ３４に通知する。もしAutoPlayであるなら、ステップＳ１０１に移行して、処理を継続する。
図５５は、プレイリスト管理テーブルにおいて”再生属性＝AutoPlay”に設定されることにより、どのような再生が行われるかを模式化した図である。ここで再生すべきタイトルは、落下するタイル片を積み重ねるというゲームアプリを含む非AV系タイトルである。この非AV系タイトルにおいて、プレイリスト管理テーブルの再生属性がAutoPlayに設定されていれば、Playback Control Engine３２によるデフォルトPL再生も開始する。ゲームアプリの実行と、デフォルトPL再生とが並列的になされるので、図５５の上段の左側に示

すように、前景をゲームアプリの画面とし、背景をデフォルトPLの再生画像とした合成画像が表示されることになる。このゲームアプリは途中で異常終了したとする。ゲームアプリはアプリケーションマネージャ３６により強制終了させられるが、デフォルトPLの再生が継続してなされるため、タイトルは、何かが写っている状態になる。このようなプレイリスト管理テーブルにおける再生属性の指定により、非AV系タイトル内のゲームアプリが異常終了した場合でも、ハングアップやブラックアウトがない動作を維持することができる。

【０１７６】

（第８実施形態）

第１実施形態においてBD-Jオブジェクトは、データ管理テーブル、アプリケーション管理テーブルという2つのテーブルを具備していたが、本実施形態は、これらを1つのテーブルに統合するという形態を開示する。かかる統合にあたって、図５６（ａ）に示すように、データ管理テーブルにおける読込属性という項目を廃し、代わりに起動属性にReady属性という属性を設ける。Ready属性とは、他のアプリケーションからの呼出又はアプリケーションマネージャ３６からの呼出に備えて、ローカルメモリ２９に予めアプリケーションをロードしておく旨を示す起動属性の類型である。

【０１７７】

図５６（ｂ）は、アプリケーションの扱いと、起動属性との関係を示した図である。第１実施形態に示したようにアプリケーションの扱いには、プリロードされるか否か(1)、現在の再生時点が有効区間に到来した際自動的に起動されるか、他からの呼出に応じて起動されるか(2)、タイトル再生進行に従ってロードされるか(3)、生存しているかという違いがあり、これらの違いにより、図５６（ｂ）に示すような5つの態様が出現する。このうち起動属性がAutoRunに設定されるのは、プリロードがなされ、”自動起動”である場合、及び、ロードがなされ、”自動起動”である場合である。

【０１７８】

一方、起動属性がReady属性に設定されるのは、プリロード、又は、ロードがなされ、起動項目が”呼出起動”を示している場合である。

尚、ワークメモリ３７では生存しているが、ローカルメモリ２９にはロードされない”との類型が存在し得ない。これは、アプリケーション・データ管理テーブルでは、ワークメモリ３７の生存区間と、ローカルメモリ２９の生存区間とが一体だからである。

【０１７９】

起動属性として、このReady属性を追加されたので、アプリケーションマネージャ３６はタイトル再生に先立ち、起動属性がAutoRunに設定されたアプリケーション、及び、起動属性がReady属性に設定されたアプリケーションをローカルメモリ２９にプリロードするとの処理を行う。こうすることにより、読込属性を設けなくても、アプリケーションをローカルメモリ２９にプリロードしておくとの処理が可能になる。

【０１８０】

図５７は、第８実施形態に係るJava仮想マシン３８によるアプリケーションの読み込みがどのようにして行われるかを模式化した図である。本図における読み込みは、図５１をベースにして作図している。

矢印1,2は、アプリケーション・データ管理テーブルに生存していて、起動属性がReady属性に設定されているJavaアーカイブファイルの読み込みを示す。

【０１８１】

矢印☆1,2,3は、アプリケーション・データ管理テーブルに生存しており、起動属性がPersistentであるアプリケーションの読み込みを示す。

これらの矢印1,2、矢印☆1,2,3は、図５１でも記述されていたものだが、図５１に記述していた、▽1,2の矢印に該当する読み込み”は、図５７では存在しない。これは、アプリケーション・データ管理テーブルは、アプリケーション管理テーブル、データ管理テーブルを一体化したものなので、アプリケーション管理テーブル＝生存、データ管理テーブル＝非存在という組合せは表現し得ないからである。

【０１８２】
以上のように本実施形態によれば、データ管理テーブル、アプリケーション管理テーブルを1つのテーブル(アプリケーション・データ管理テーブル)にまとめることができるので、アプリケーションマネージャ３６による処理を簡略化することができる。尚、読込優先度をなくすことによりアプリケーション・データ管理テーブルをより簡略化にしても良い。
【０１８３】
（第９実施形態）
第１実施形態では、アプリケーションをローカルメモリ２９に読み込むにあたって、読込優先度を参照して、この読込優先度に従い、読み込み処理に優劣を与えた。これに対し第９実施形態は、Optionalを意味する情報と、0から255までの数値との組合せにより読込優先度を表す実施形態である。
【０１８４】
図５８（ａ）（ｂ）は、第９実施形態に係る読込優先度の一例を示す図である。255、128は、0から255までの読込優先度の一例であり、本例におけるapplication#2は、application#3より読込優先度が高いことを意味する。
本実施形態においてアプリケーションマネージャ３６は、第１実施形態同様、先ずMandatoryを示す読込優先度が付与されたアプリケーションをローカルメモリ２９に読み込む。
【０１８５】
その後、Optionalを示す読込優先度が付与されたアプリケーションに対しては、ローカルメモリ２９における容量が、アプリケーションのサイズを上回るか否かを判定する。もし上回るなら、読込優先度＝Optionalが付与されたアプリケーションをそのままローカルメモリ２９に読み込む。もし下回るなら、アプリケーションを構成するデータのうち、読込優先度を表す数値が高いアプリケーションをローカルメモリ２９に読み込む。そして、ローカルメモリ２９における残りの領域に、読込優先度を表す数値が低いアプリケーションを読み出す。
【０１８６】
こうすることでOptional扱いのアプリケーションについては、全体を格納する容量が再生装置のローカルメモリ２９になくても、その一部分をローカルメモリ２９に格納しておくことができる。
（第１０実施形態）
第１実施形態においてアプリケーションマネージャ３６は、同じapplicationIDが付与されたアプリケーションを、読込優先度に従い排他的にローカルメモリ２９にロードするとしたが、第１０実施形態は、アプリケーションにグループ属性を与えることにより、排他的なロードを実現する。図５９は、グループ属性が付与されたデータ管理テーブルを示す図である。グループ属性には、排他グループなし、排他グループあり、といった、2通りの設定が可能であり、排他グループありの場合、そのグループ番号が記述される。図５９（ａ）におけるapplication#1の「－」は、排他グループが存在しないことを示す。一方、application#2,#3の「group#1」は、排他グループがあり、application#2,#3は、group#1という排他グループに帰属していることを示す。以上が本実施形態に係る記録媒体の改良である。
【０１８７】
本実施形態に係る再生装置は、データ管理テーブルに基づいて各アプリケーションをローカルメモリ２９に読み込んだ後、ローカルメモリ２９のアプリケーションにおけるグループ属性をベリファイする。同じ排他グループに帰属するアプリケーションが、ローカルメモリ２９上に2つ以上存在していれば、そのうち一方をローカルメモリ２９から削除する。
【０１８８】
こうすることにより、ローカルメモリ２９の利用効率を向上させることができる。排他

グループの具体例としては、ランチャーアプリと、このアプリにより起動されるアプリとからなるグループが相応しい。本アプリケーションにより起動されるアプリケーションは、原則1つに限られるので、ローカルメモリ２９には、ランチャー＋１個のアプリケーションのみが存在する筈である。もし3つ以上のアプリケーションが存在していれば、これをローカルメモリ２９から削除するという処理をアプリケーションマネージャ３６は行う必要があるので、各アプリケーションのグループ属性を設け、ローカルメモリ２９上で存在するアプリケーションがランチャー＋１個のアプリケーションになっているかどうかのチェックを行うのである。

【０１８９】

図５９（ａ）は、アプリケーション管理テーブルに基づくローカルメモリ２９に対するアクセスを示す図である。本図において、読込優先度＝Optionalと設定されたapplication#2、application#3のグループ属性は、group#1であるので、これらのアプリケーションは、同じ排他グループに属することになる。3つのアプリケーションのうち、application#1は上述したランチャーアプリケーションであり、application#2、application#3は、これにより起動されるアプリケーションであるので、どちらかのみがローカルメモリ２９上に存在するよう、グループ属性が付与されている。アプリケーションマネージャ３６は、これらapplication#2、application#3のグループ属性を参照して、どちらか1つをローカルメモリ２９から削除するとの処理を行う。かかる削除によりローカルメモリ２９に余白が生まれる。

【０１９０】

（第１１実施形態）

第１実施形態では、アプリケーション管理テーブルをタイトル毎に持たせるとしたが、本実施形態では、このアプリケーション管理テーブルの割当単位を変更させることを提案する。図６０は、割当単位のバリエーションを示す図である。本図において第1段目は、BD-ROMに記録されている3つのアプリケーション管理テーブルを示し、第2段目は、タイトル単位、第3段目は、ディスク単位、第4段目は、複数BD-ROMからなるディスクセット単位を示す。図中の矢印は、アプリケーション管理テーブルの割り当てを模式化して示している。この矢印を参照すると、第1段目におけるアプリケーション管理テーブル#1,#2,#3のそれぞれは、第2段目に示したtitle#1,#2,#3のそれぞれに割り当てられていることがわかる。また、ディスク単位ではアプリケーション管理テーブル#4が割り当てられており、ディスクセット全体に対しはアプリケーション管理テーブル#5が割り当てられている。このようにアプリケーション管理テーブルの割当単位を、タイトルより大きい単位にすることにより、1つのBD-ROMがローディングされている間、生存するようなアプリケーションや複数BD-ROMのうちどれかがローディングされている間、生存するようなアプリケーションを定義することができる。

（備考）

以上の説明は、本発明の全ての実施行為の形態を示している訳ではない。下記(A)(B)(C)(D)・・・・・の変更を施した実施行為の形態によっても、本発明の実施は可能となる。本願の請求項に係る各発明は、以上に記載した複数の実施形態及びそれらの変形形態を拡張した記載、ないし、一般化した記載としている。拡張ないし一般化の程度は、本発明の【技術分野】の、出願当時の技術水準の特性に基づく。

【０１９１】

(A)全ての実施形態では、本発明に係る光ディスクをBD-ROMとして実施したが、本発明の光ディスクは、記録される動的シナリオ、Index Tableに特徴があり、この特徴は、BD-ROMの物理的性質に依存するものではない。動的シナリオ、Index Tableを記録しうる記録媒体なら、どのような記録媒体であってもよい。例えば、DVD-ROM,DVD-RAM,DVD-RW,DVD-R,DVD+RW,DVD+R,CD-R,CD-RW等の光ディスク、PD,MO等の光磁気ディスクであってもよい。また、コンパクトフラッシュカード、スマートメディア、メモリスティック、マルチメディアカード、PCM-CIAカード等の半導体メモリカードであってもよい。フレシキブルディスク、SuperDisk,Zip,Clik!等の磁気記録ディスク(i)、ORB,Jaz,SparQ,SyJet,EZFley,マ

イクロドライブ等のリムーバルハードディスクドライブ(ii)であってもよい。更に、機器内蔵型のハードディスクであってもよい。

(B) 全ての実施形態における再生装置は、BD-ROMに記録されたAVClipをデコードした上でTVに出力していたが、再生装置をBD-ROMドライブのみとし、これ以外の構成要素をTVに具備させてもい、この場合、再生装置と、TVとをIEEE1394で接続されたホームネットワークに組み入れることができる。また、実施形態における再生装置は、テレビと接続して利用されるタイプであったが、ディスプレィと一体型となった再生装置であってもよい。更に、各実施形態の再生装置において、処理の本質的部分をなす部分のみを、再生装置としてもよい。これらの再生装置は、何れも本願明細書に記載された発明であるから、これらの何れの態様であろうとも、各実施形態に示した再生装置の内部構成を元に、再生装置を製造する行為は、本願の明細書に記載された発明の実施行為になる。各実施形態に示した再生装置の有償・無償による譲渡(有償の場合は販売、無償の場合は贈与になる)、貸与、輸入する行為も、本発明の実施行為である。店頭展示、カタログ勧誘、パンフレット配布により、これらの譲渡や貸渡を、一般ユーザに申し出る行為も本再生装置の実施行為である。

【０１９２】

(C)各フローチャートに示したプログラムによる情報処理は、ハードウェア資源を用いて具体的に実現されていることから、上記フローチャートに処理手順を示したプログラムは、単体で発明として成立する。全ての実施形態は、再生装置に組み込まれた態様で、本発明に係るプログラムの実施行為についての実施形態を示したが、再生装置から分離して、各実施形態に示したプログラム単体を実施してもよい。プログラム単体の実施行為には、これらのプログラムを生産する行為(1)や、有償・無償によりプログラムを譲渡する行為(2)、貸与する行為(3)、輸入する行為(4)、双方向の電子通信回線を介して公衆に提供する行為(5)、店頭展示、カタログ勧誘、パンフレット配布により、プログラムの譲渡や貸渡を、一般ユーザに申し出る行為(6)がある。

【０１９３】

(D)各フローチャートにおいて時系列に実行される各ステップの「時」の要素を、発明を特定するための必須の事項と考える。そうすると、これらのフローチャートによる処理手順は、再生方法の使用形態を開示していることがわかる。各ステップの処理を、時系列に行うことで、本発明の本来の目的を達成し、作用及び効果を奏するよう、これらのフローチャートの処理を行うのであれば、本発明に係る記録方法の実施行為に該当することはいうまでもない。

【０１９４】

(E)Chapterを一覧表示するためのMenu(Chapter Menu)と、これの挙動を制御するMOVIEオブジェクトとをBD-ROMに記録しておき、Top Menuから分岐できるようにしてもよい。またリモコンキーのChapterキーの押下により呼出されるようにしてもよい。

(F)BD-ROMに記録するにあたって、AVClipを構成する各TSパケットには、拡張ヘッダを付与しておくことが望ましい。拡張ヘッダは、TP_extra_headerと呼ばれ、『Arrival_Time_Stamp』と、『copy_permission_indicator』とを含み4バイトのデータ長を有する。TP_extra_header付きTSパケット(以下EX付きTSパケットと略す)は、32個毎にグループ化されて、3つのセクタに書き込まれる。32個のEX付きTSパケットからなるグループは、6144バイト(=32×192)であり、これは3個のセクタサイズ6144バイト(=2048×3)と一致する。3個のセクタに収められた32個のEX付きTSパケットを”Aligned Unit”という。

【０１９５】

IEEE1394を介して接続されたホームネットワークでの利用時において、再生装置２００は、以下のような送信処理にてAligned Unitの送信を行う。つまり送り手側の機器は、Aligned Unitに含まれる32個のEX付きTSパケットのそれぞれからTP_extra_headerを取り外し、TSパケット本体をDTCP規格に基づき暗号化して出力する。TSパケットの出力にあたっては、TSパケット間の随所に、isochronousパケットを挿入する。この挿入箇所は、TP_extra_headerのArrival_Time_Stampに示される時刻に基づいた位置である。TSパケットの出

力に伴い、再生装置２００はDTCP_Descriptorを出力する。DTCP_Descriptorは、TP_extra_headerにおけるコピー許否設定を示す。ここで「コピー禁止」を示すようDTCP_Descriptorを記述しておけば、IEEE1394を介して接続されたホームネットワークでの利用時においてTSパケットは、他の機器に記録されることはない。

【０１９６】

(G)各実施形態において、記録媒体に記録されるデジタルストリームはAVClipであったが、DVD-Video規格、DVD-Video Recording規格のVOB(Video Object)であってもよい。VOBは、ビデオストリーム、オーディオストリームを多重化することにより得られたISO/IEC13818-1規格準拠のプログラムストリームである。またAVClipにおけるビデオストリームは、MPEG4やWMV方式であってもよい。更にオーディオストリームは、Linear-PCM方式、Dolby-AC3方式、MP3方式、MPEG-AAC方式、Dts、WMA(Windows media audio)であってもよい。

【０１９７】

(H)各実施形態における映像作品は、アナログ放送で放送されたアナログ映像信号をエンコードすることにより得られたものでもよい。デジタル放送で放送されたトランスポートストリームから構成されるストリームデータであってもよい。

またビデオテープに記録されているアナログ／デジタルの映像信号をエンコードしてコンテンツを得ても良い。更にビデオカメラから直接取り込んだアナログ／デジタルの映像信号をエンコードしてコンテンツを得ても良い。他にも、配信サーバにより配信されるデジタル著作物でもよい。

【０１９８】

(I)BD-Jモジュール３５は、衛星放送受信のために機器に組み込まれたJavaプラットフォームであってもよい。BD-Jモジュール３５がかかるJavaプラットフォームであれば、本発明に係る再生装置は、MHP用STBとしての処理を兼用することになる。

更に携帯電話の処理制御のために機器に組み込まれたJavaプラットフォームであってもよい。かかるBD-Jモジュール３５がかかるJavaプラットフォームであれば、本発明に係る再生装置は、携帯電話としての処理を兼用することになる。

【０１９９】

(J)レイアモデルにおいて、BD-Jモードの上にMOVIEモードを配置してもよい。特にMOVIEモードでの動的シナリオの解釈や、動的シナリオに基づく制御手順の実行は、再生装置に対する負担が軽いので、MOVIEモードをBD-Jモード上で実行させても何等問題は生じないからである。また再生装置や映画作品の開発にあたって、動作保証が1つのモードで済むからである。

【０２００】

更にBD-Jモードだけで再生処理を実行してもよい。第５実施形態に示したように、BD-JモードでもPLの再生と同期した再生制御が可能になるから、強いてMOVIEモードを設けなくてもよいという理由による。

(K)AVClipに多重化されるべきインタラクティブグラフィクスストリームにナビゲーションコマンドを設けて、あるPLから別のPLへの分岐を実現しても良い。

【産業上の利用可能性】

【０２０１】

本発明に係る再生装置は、ホームシアターシステムでの利用のように、個人的な用途で利用されることがありうる。しかし本発明は上記実施形態に内部構成が開示されており、この内部構成に基づき量産することが明らかなので、資質において工業上利用することができる。このことから本発明に係る再生装置は、産業上の利用可能性を有する。

【図面の簡単な説明】

【０２０２】

【図１】本発明に係る再生装置の使用行為についての形態を示す図である。

【図２】BD-ROMにおけるファイル・ディレクトリ構成を示す図である。

【図３】AVClip時間軸と、PL時間軸との関係を示す図である。

【図４】4つのClip_Information_file_nameによりなされた一括指定を示す図である。

【図５】PLmarkによるチャプター定義を示す図である。
【図６】SubPlayItem時間軸上の再生区間定義と、同期指定とを示す図である。
【図７】　（ａ）Movieオブジェクトの内部構成を示す図である。
（ｂ）BD-Jオブジェクトの内部構成を示す図である。
（ｃ）Javaアプリケーションの内部構成を示す図である。
【図８】　（ａ）Javaアーカイブファイルに収められているプログラム、データを示す図である。
（ｂ）xletプログラムの一例を示す図である。
【図９】　（ａ）トップメニュー、title#1、title#2といった一連のタイトルを示す図である。
（ｂ）PlayList#1、PlayList#2の時間軸を足し合わせた時間軸を示す図である。
【図１０】本編タイトル、オンラインショッピングタイトル、ゲームタイトルという3つのタイトルを含むディスクコンテンツを示す図である。
【図１１】図１０に示した3つのタイトルの再生画像の一例を示す図である。
【図１２】　（ａ）図１０の破線に示される帰属関係から各アプリケーションの生存区間をグラフ化した図である。
（ｂ）図１２（ａ）の生存区間を規定するため、記述されたアプリケーション管理テーブルの一例を示す図である。
【図１３】　（ａ）起動属性設定の一例を示す図である。
（ｂ）他のアプリケーションからのアプリケーション呼出があって初めて起動するアプリケーション(application#2)を示す図である。
【図１４】　（ａ）（ｂ）Suspendが有意義となるアプリケーション管理テーブル、生存区間の一例を示す図である。
【図１５】起動属性がとり得る三態様(Persistent、AutoRun、Suspend)と、直前タイトルにおけるアプリケーション状態の三態様(非起動、起動中、Suspend)とがとりうる組合せを示す図である。
【図１６】本発明に係る再生装置の内部構成を示す図である。
【図１７】　（ａ）BD-ROMに存在しているJavaアーカイブファイルを、ローカルメモリ２９上でどのように識別するかを示す図である。
（ｂ）図１７（ａ）の応用を示す図である。
【図１８】ROM２４に格納されたソフトウェアと、ハードウェアとからなる部分を、レイア構成に置き換えて描いた図である。
【図１９】Presentation Engine３１～モジュールマネージャ３４による処理を模式化した図である。
【図２０】アプリケーションマネージャ３６による処理を模式化した図である。
【図２１】ワークメモリ３７～Default Operation Manager４０を示す図である。
【図２２】アプリケーションマネージャ３６による分岐時の制御手順を示す図である。
【図２３】アプリケーション終了処理の処理手順を示すフローチャートである。
【図２４】アプリケーション終了の過程を模式的に示した図である。
【図２５】　（ａ）PL時間軸上に生存区間を定めたアプリケーション管理テーブルを示す図である。
（ｂ）図２５（ａ）のアプリケーション管理テーブルに基づき、アプリケーションの生存区間を示した図である。
【図２６】（ａ）PL時間軸から定まるタイトル時間軸を示す。
（ｂ）メインとなるアプリケーションの生存区間から定まるタイトル時間軸を示す。
（ｃ）複数アプリケーションの生存区間から定まるタイトル時間軸を示す図である。
【図２７】タイトル再生時におけるアプリケーションマネージャ３６の処理手順を示すフローチャートである。
【図２８】　（ａ）BD-ROMにより実現されるメニュー階層を示す図である。
（ｂ）メニュー階層を実現するためのMOVIEオブジェクトを示す図である。

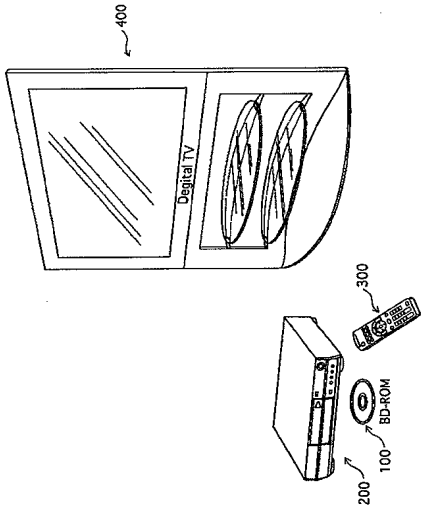
【図２９】Index Tableと、Index Tableから各Movieオブジェクトへの分岐とを模式化した図である。
【図３０】　（ａ）図２９（ｂ）のようにIndex Tableが記述された場合における分岐を示す。
（ｂ）非AV系タイトルが強制終了した際における分岐を示す図である。
【図３１】モジュールマネージャ３４の処理手順を示すフローチャートである。
【図３２】アプリケーションマネージャ３６によるアプリケーション強制終了の動作例を示す図である。
【図３３】Playback Control Engine３２によるPL再生手順を示すフローチャートである。
【図３４】アングル切換、SkipBack,SkipNextの受付手順を示すフローチャートである。
【図３５】SkipBack,SkipNextAPIがコールされた際の処理手順を示すフローチャートである。
【図３６】Presentation Engine３１による処理手順の詳細を示すフローチャートである。
【図３７】SubPlayItemの再生手順を示すフローチャートである。
【図３８】第５実施形態に係るアプリケーションマネージャ３６の処理手順を示すフローチャートである。
【図３９】データ管理テーブルの一例を示す図である。
【図４０】BD-Jオブジェクトが想定している実行モデルを示す図である。
【図４１】　（ａ）ローカルメモリ２９におけるJavaアーカイブファイル生存を示す生存区間を示す図である。
（ｂ）図４１（ａ）でのJavaアーカイブファイル生存区間を規定するため、記述されたデータ管理テーブルを示す図である。
【図４２】カルーセル化によるJavaアーカイブファイル埋め込みを示す図である。
【図４３】　（ａ）インターリーブ化によるAVClip埋め込みを示す図である。
（ｂ）読込属性の3つの類型を示す図である。
【図４４】　（ａ）データ管理テーブルの一例を示す図である。
（ｂ）図４４（ａ）のデータ管理テーブルの割り当てによるローカルメモリ２９の格納内容の変遷を示す図である。
【図４５】　（ａ）新旧再生装置におけるローカルメモリ２９のメモリ規模を対比して示す図である。
（ｂ）読込優先度が設定されたデータ管理テーブルの一例を示す図である。
【図４６】アプリケーションマネージャ３６によるプリロード制御の処理手順を示す図である。
【図４７】　（ａ）applicationIDが同一であるが、読込優先度は互いに異なる複数のアプリケーションを規定するデータ管理テーブルの一例を示す図である。
（ｂ）図４７（ａ）のデータ管理テーブルの割り当てによるローカルメモリ２９の格納内容の変遷を示す図である。
【図４８】　（ａ）プリロードされるべきアプリケーション、ロードされるべきアプリケーションに同一のapplicationIDを付与するよう記述されたデータ管理テーブルの一例を示す図である。
（ｂ）メモリ規模が小さい再生装置におけるローカルメモリ２９の格納内容の変遷を示す図である。
（ｃ）メモリ規模が大きい再生装置におけるローカルメモリ２９の格納内容の変遷を示す図である。
【図４９】データ管理テーブルに基づくアプリケーションマネージャ３６によるロード処理の処理手順を示す図である。
【図５０】アプリケーションqの生存区間に、現在の再生時点が到達した場合のアプリケーションマネージャ３６による処理手順を示す図である。

【図５１】Java仮想マシン３８によるアプリケーションの読み込みがどのようにして行われるかを模式化した図である。
【図５２】　（ａ）第７実施形態に係るBD-Jオブジェクトの内部構成を示す図である。
（ｂ）プレイリスト管理テーブルの一例を示す図である。
（ｃ）分岐先タイトルのプレイリスト管理テーブルにおいて、再生属性がAutoPlayに設定されたPLが存在する場合、再生装置がどのような処理を行うかを示す図である。
【図５３】　（ａ）再生属性が非自動再生を示すよう設定された場合の非AV系タイトルにおけるタイトル時間軸を示す図である。
（ｂ）再生属性がAutoPlayに設定された非AV系タイトルのタイトル時間軸を示す図である。
（ｃ）プレイリスト管理テーブルにおいて再生属性が”AutoPlay”を示すよう設定され、アプリケーションが強制終了した場合を示す図である。
（ｄ）プレイリスト管理テーブルにおいて再生属性が”AutoPlay”を示すよう設定され、メインアプリの起動に失敗したケースを示す図である。
【図５４】第７実施形態に係るアプリケーションマネージャ３６の処理手順を示すフローチャートである。
【図５５】プレイリスト管理テーブルにおいて”再生属性＝AutoPlay”に設定されることにより、どのような再生が行われるかを模式化した図である。
【図５６】　（ａ）（ｂ）アプリケーションの扱いと、起動属性との関係を示した図である。
【図５７】第８実施形態に係るJava仮想マシン３８によるアプリケーションの読み込みがどのようにして行われるかを模式化した図である。
【図５８】（ａ）（ｂ）第９実施形態に係る読込優先度の一例を示す図である。
【図５９】　（ａ）グループ属性が付与されたデータ管理テーブルを示す図である。
（ｂ）アプリケーション管理テーブルに基づくローカルメモリ２９に対するアクセスを示す図である。
【図６０】アプリケーション管理テーブルの割当単位のバリエーションを示す図である。
【符号の説明】
【０２０３】

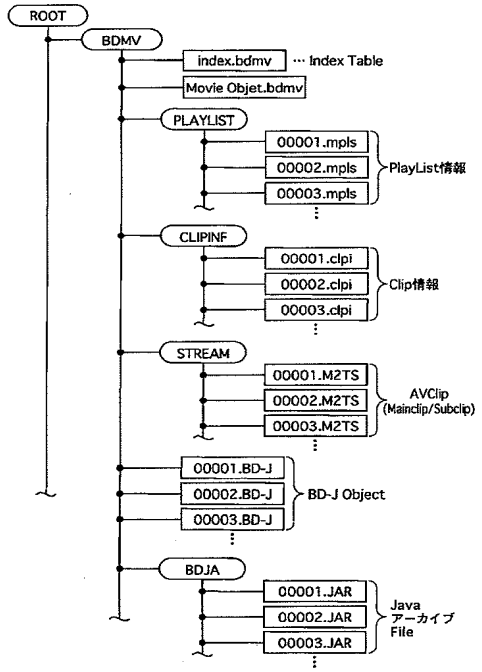
１	BD-ROMドライブ
２	リードバッファ
３	デマルチプレクサ
４	ビデオデコーダ
５	ビデオプレーン
９	P-Graphicsデコーダ
１０	Presentation Graphicsプレーン
１１	合成部
１２	フォントゼネレータ
１３	I-Graphicsデコーダ
１４	スイッチ
１５	Interactive Graphicsプレーン
１６	合成部
１７	HDD
１８	リードバッファ
１９	デマルチプレクサ
２０	オーディオデコーダ
２１	シナリオメモリ
２２	CPU
２３	キーイベント処理部
２４	命令ROM

２５　スイッチ
２６　CLUT部
２７　CLUT部
２８　PSRセット
２９　ローカルメモリ
３１　Presentation Engine
３２　再生制御エンジン
３３　HDMVモジュール
３４　モジュールマネージャ
３６　アプリケーションマネージャ
３７　ワークメモリ
３８　Java仮想マシン
３９　Event Listner Manager
４０　Default Operation Manage

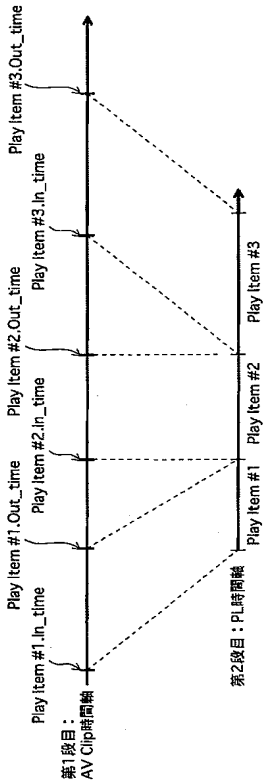
【図１】



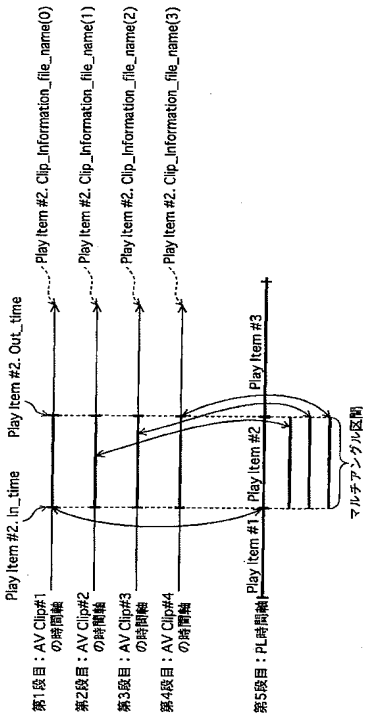
【図２】



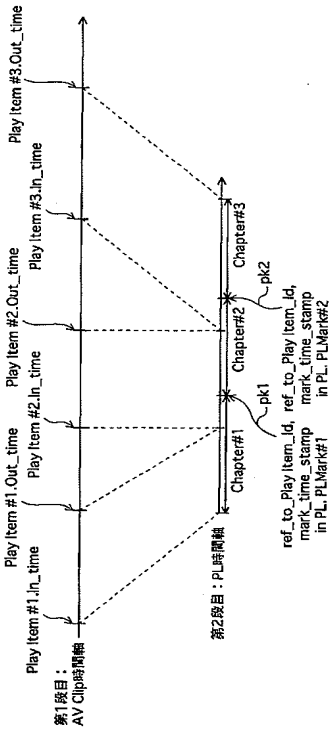
【図３】


Play Item #1.In_time
Play Item #1.Out_time
Play Item #2.In_time
Play Item #2.Out_time
Play Item #3.In_time
Play Item #3.Out_time
第1段目：
AV Clip時間軸
第2段目：PL時間軸
Play Item #1
Play Item #2
Play Item #3

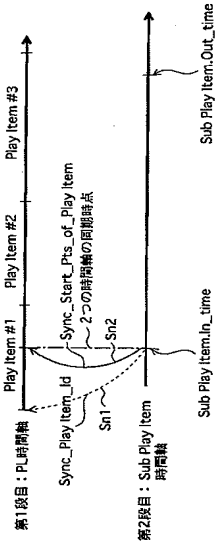
【図４】


Play Item #2. In_time
Play Item #2. Out_time
Play Item #2. Clip_Information_file_name(0)
Play Item #2. Clip_Information_file_name(1)
Play Item #2. Clip_Information_file_name(2)
Play Item #2. Clip_Information_file_name(3)
第1段目：AV Clip#1の時間軸
第2段目：AV Clip#2の時間軸
第3段目：AV Clip#3の時間軸
第4段目：AV Clip#4の時間軸
第5段目：PL時間軸
Play Item #1
Play Item #2
Play Item #3
マルチアングル区間

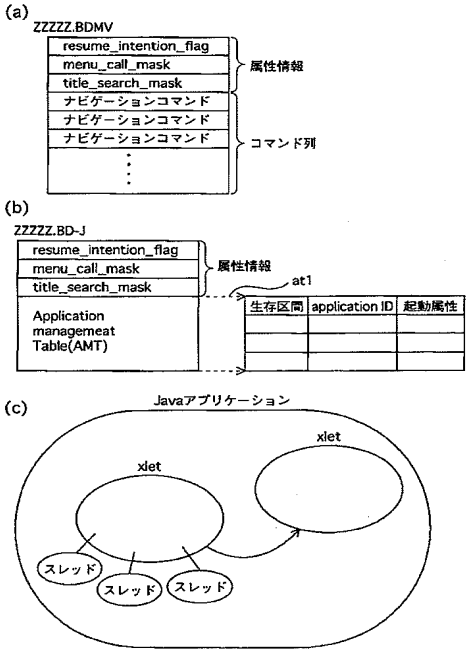
【図５】


Play Item #1.In_time
Play Item #1.Out_time
Play Item #2.In_time
Play Item #2.Out_time
Play Item #3.In_time
Play Item #3.Out_time
第1段目：
AV Clip時間軸
第2段目：PL時間軸
Chapter#1
Chapter#2
Chapter#3
pk1
pk2
ref_to_Play Item_Id,
mark_time_stamp
in PL_PLMark#1
ref_to_Play Item_Id,
mark_time_stamp
in PL_PLMark#2

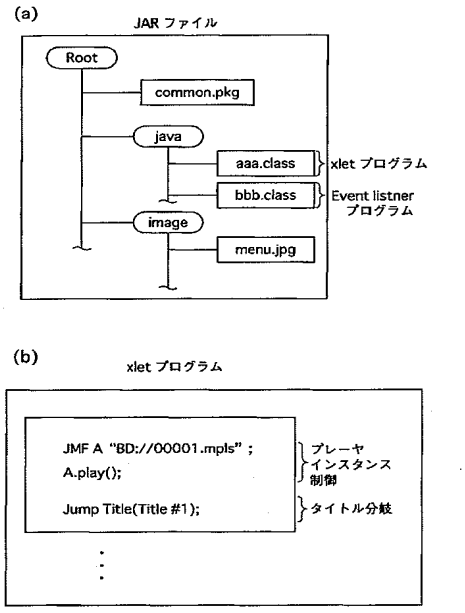
【図６】


第1段目：PL時間軸
Play Item #1
Play Item #2
Play Item #3
Sync_Play Item_Id
Sync_Start_Pts_of_Play Item
2つの時間軸の同期時点
Sn1
Sn2
第2段目：Sub Play Item
時間軸
Sub Play Item.In_time
Sub Play Item.Out_time

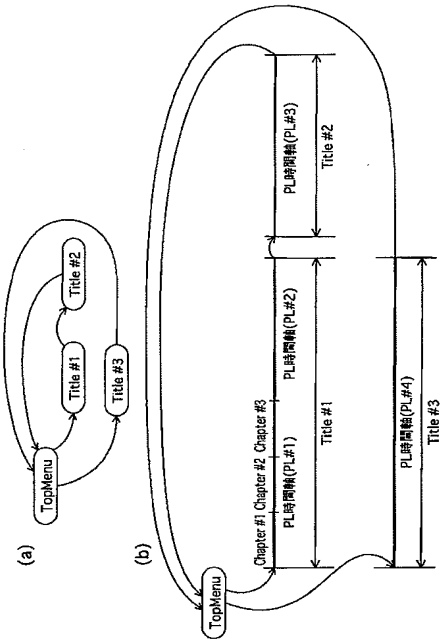
【図７】


(a)
ZZZZZ.BDMV
resume_intention_flag
menu_call_mask
title_search_mask
ナビゲーションコマンド
ナビゲーションコマンド
ナビゲーションコマンド
属性情報
コマンド列
(b)
ZZZZZ.BD-J
resume_intention_flag
menu_call_mask
title_search_mask
属性情報
at1
Application managemeat Table(AMT)
生存区間
application ID
起動属性
(c)
Javaアプリケーション
xlet
xlet
スレッド
スレッド
スレッド

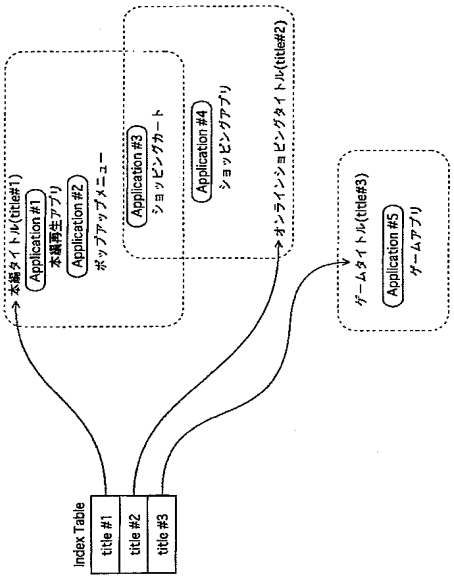
【図８】


(a)
JARファイル
Root
common.pkg
java
aaa.class
bbb.class
image
menu.jpg
xletプログラム
Event listner プログラム
(b)
xletプログラム
JMF A "BD://00001.mpls" ;
A.play();
Jump Title(Title #1);
プレーヤインスタンス制御
タイトル分岐

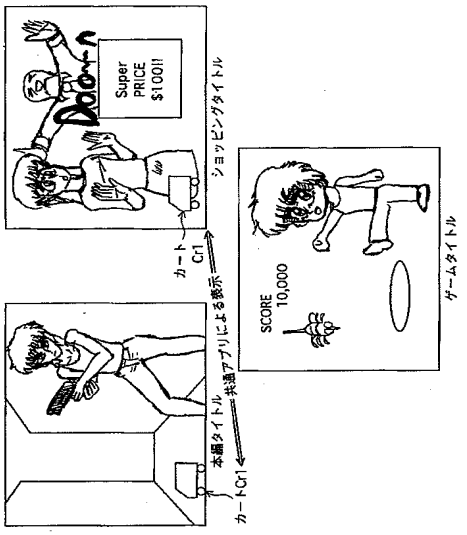
【図９】


(a)
TopMenu
Title #1
Title #2
Title #3
(b)
TopMenu
Chapter #1
Chapter #2
Chapter #3
PL時間軸(PL#1)
PL時間軸(PL#2)
PL時間軸(PL#3)
PL時間軸(PL#4)
Title #1
Title #2
Title #3

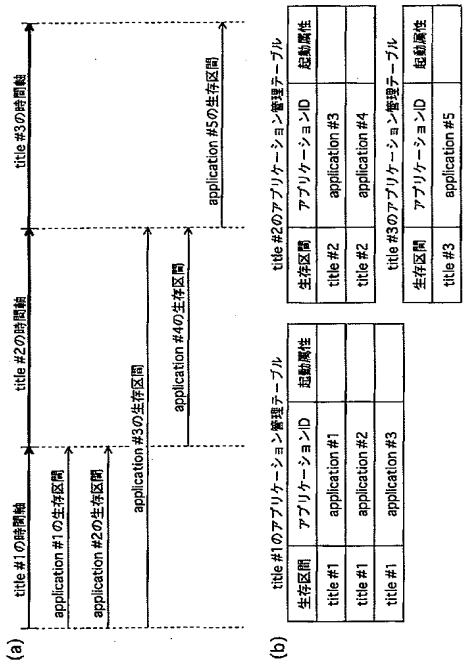
【図１０】


本編タイトル(title#1)
Application #1
本編再生アプリ
Application #2
ポップアップメニュー
Application #3
ショッピングカート
Application #4
ショッピングアプリ
オンラインショッピングタイトル(title#2)
ゲームタイトル(title#3)
Application #5
ゲームアプリ
Index Table
title #1
title #2
title #3

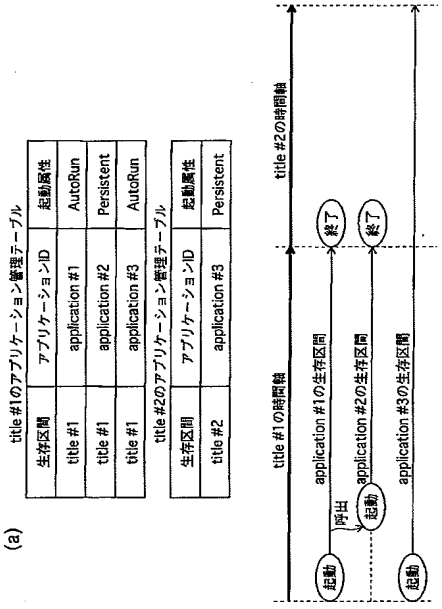
【図１１】



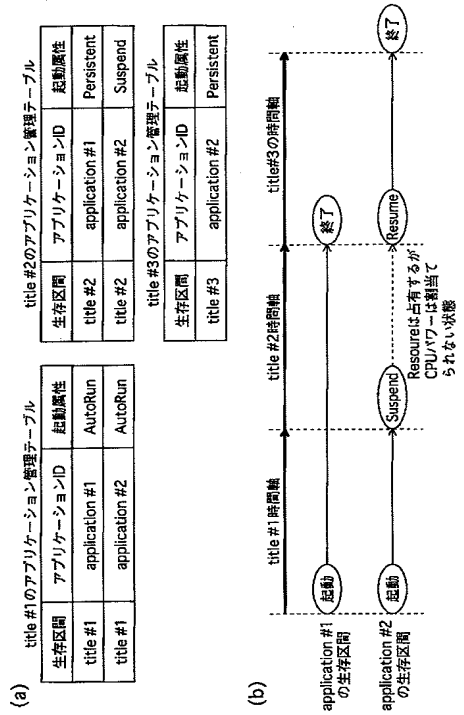
【図１２】



【図１３】



【図１４】

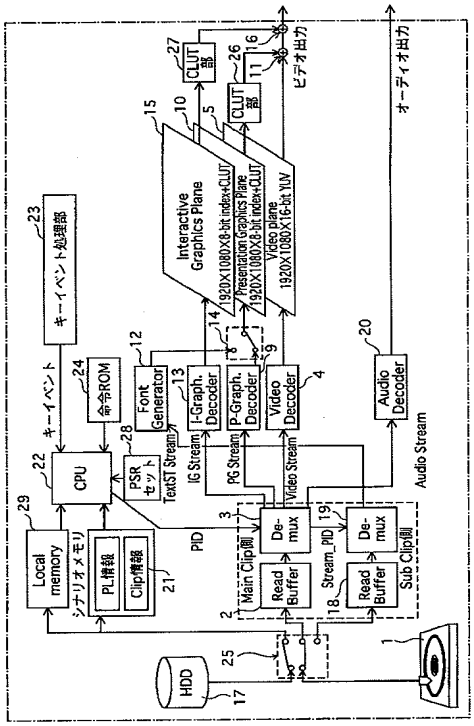


【図１５】

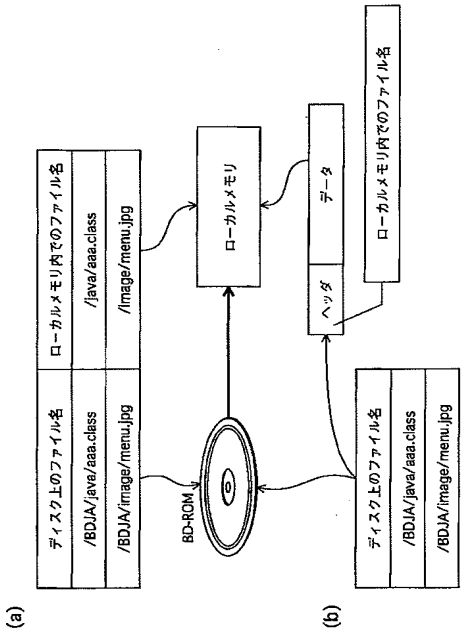
起動属性によるアプリケーション状態の変化

直前タイトルにおけるアプリケーションの状態	起動属性			
	非起動	起動中	Suspend	
Persistent	何もせず、状態継続	何もせず、状態継続	レジュームする	
AutoRun	アプリケーションを起動する	何もせず、状態継続	レジュームする	
Suspend	何もせず、状態継続	サスペンドする	何もせず、状態継続	

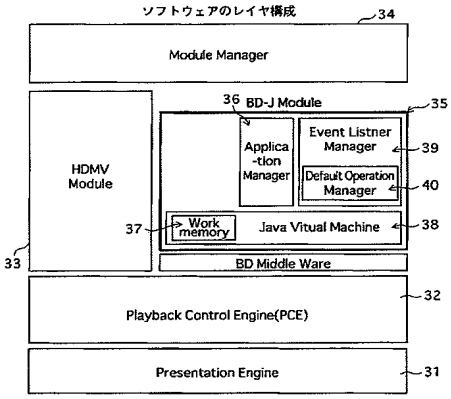
【図１６】



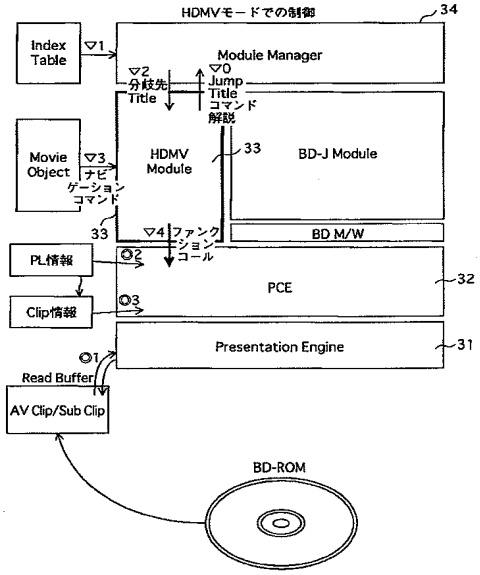
【図１７】



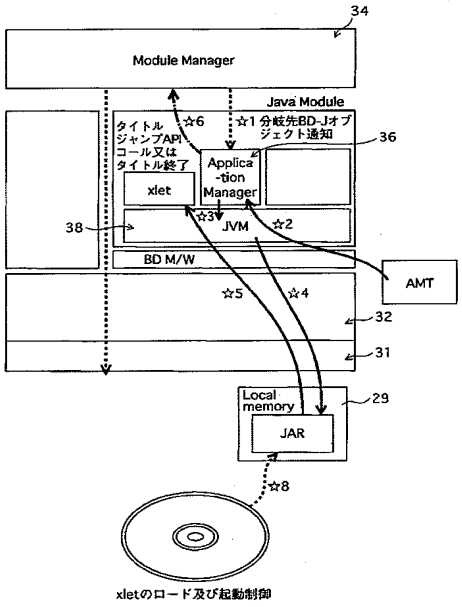
【図１８】



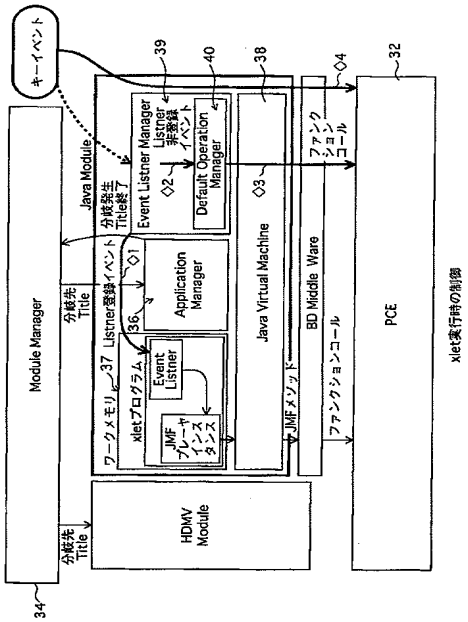
【図１９】



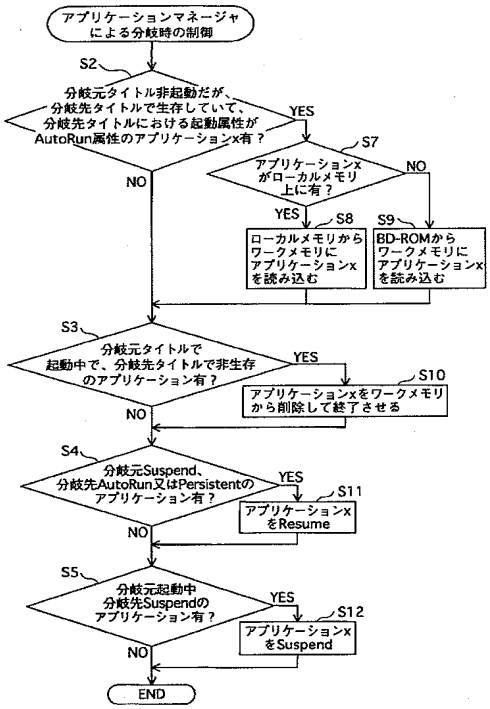
【図２０】



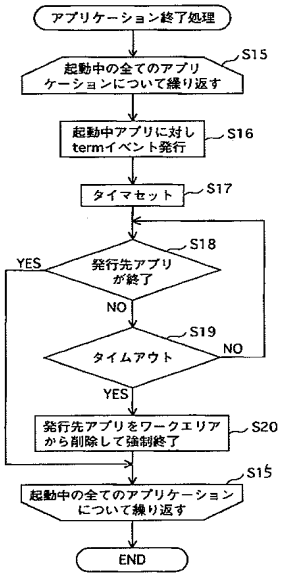
【図２１】



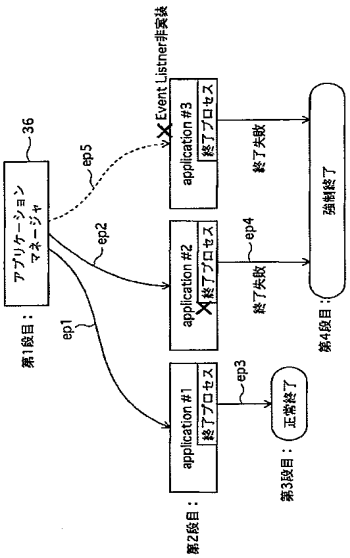
【図２２】



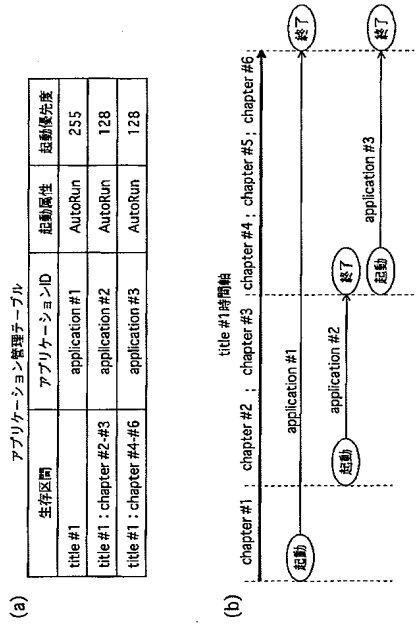
【図２３】



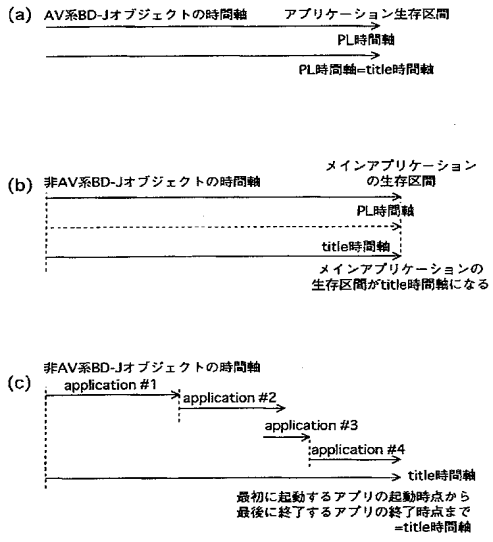
【図２４】



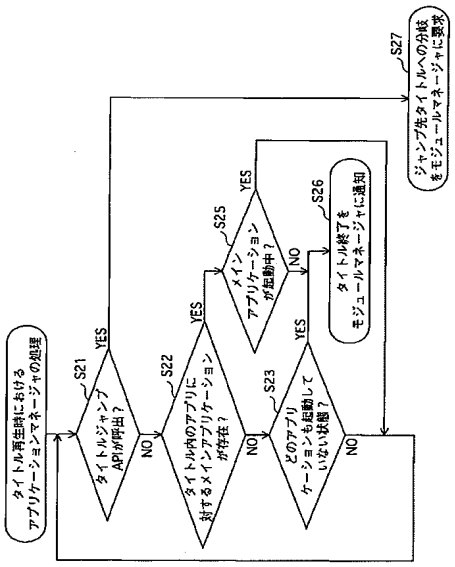
【図２５】



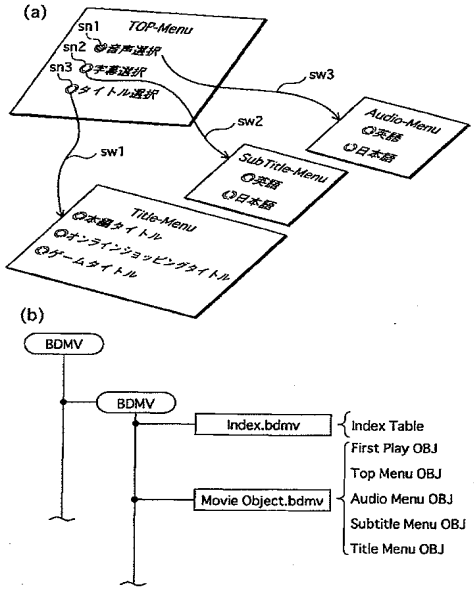
【図２６】



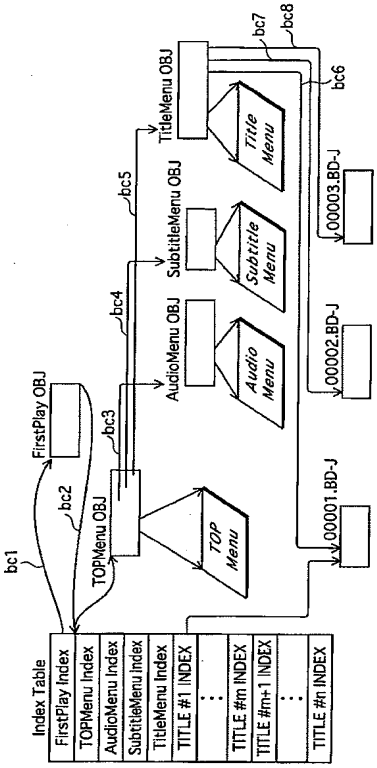
【図２７】



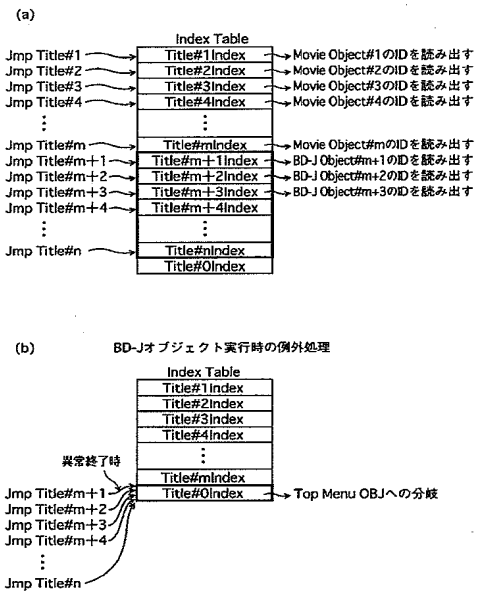
【図２８】



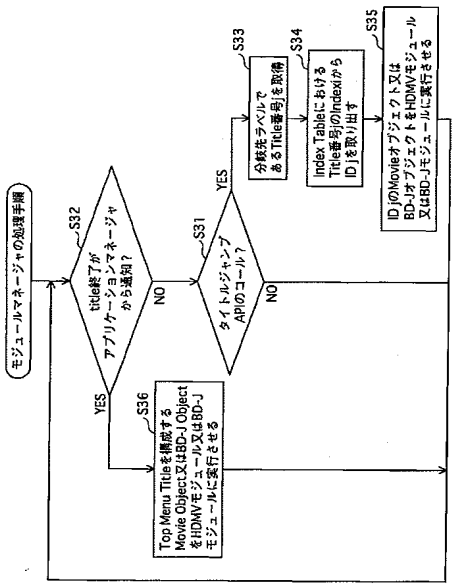
【図２９】



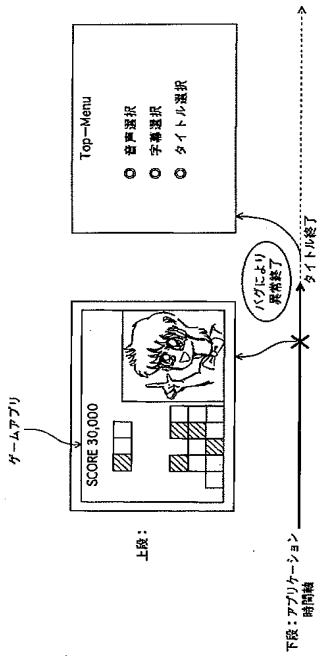
【図３０】



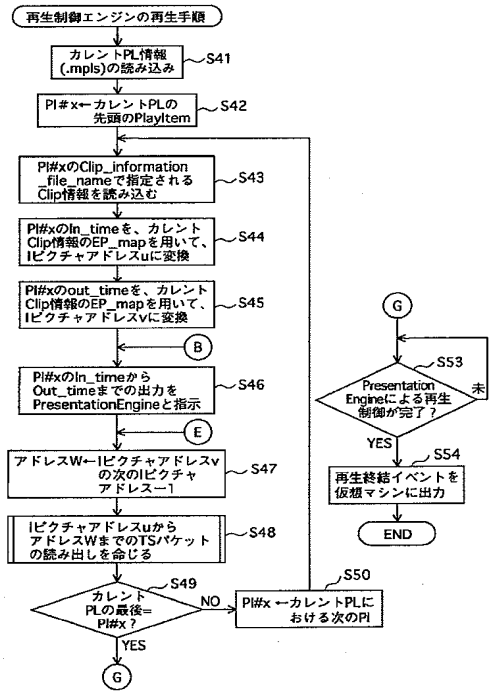
【図３１】



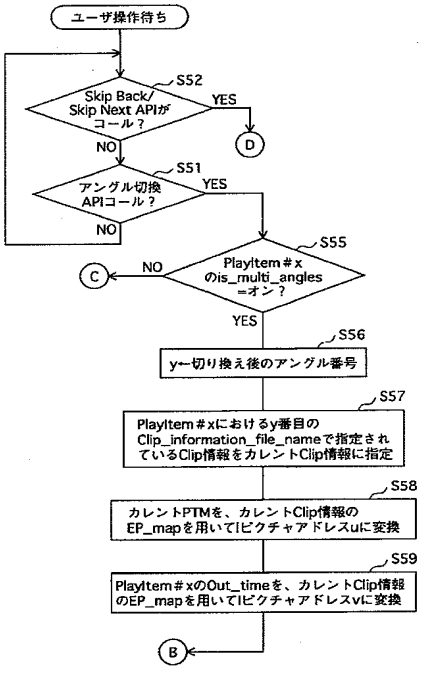
【図３２】



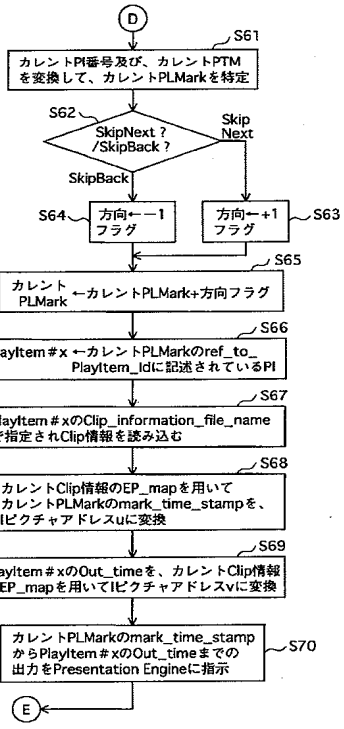
【図３３】



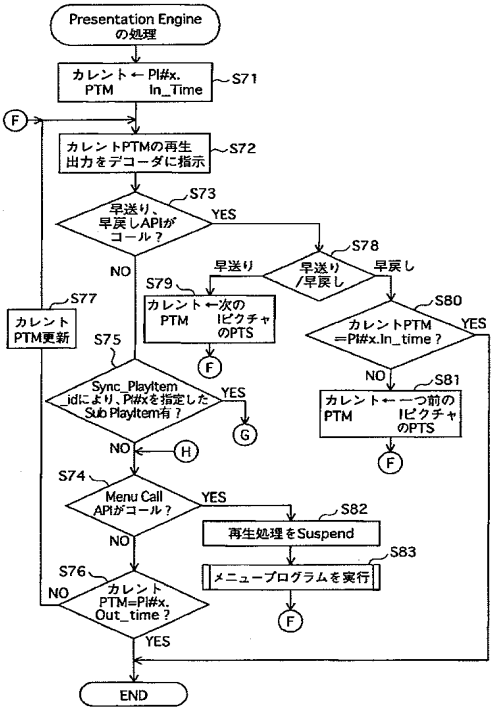
【図３４】



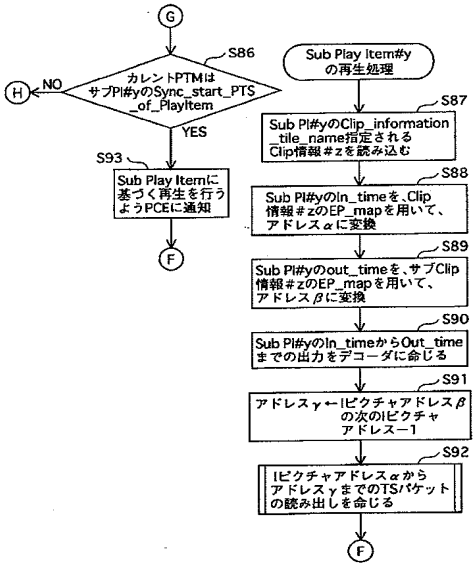
【図３５】



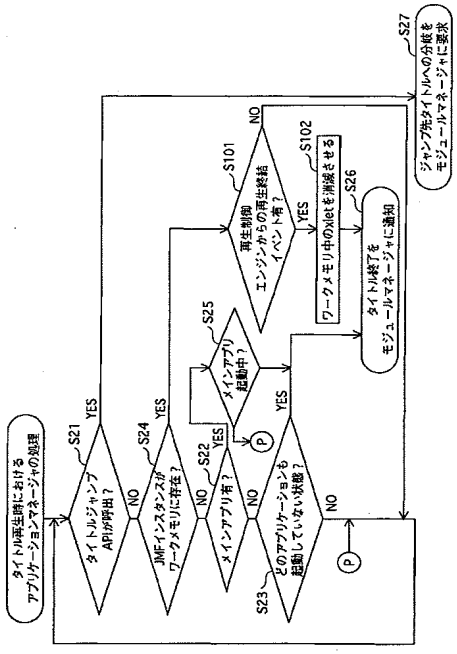
【図３６】



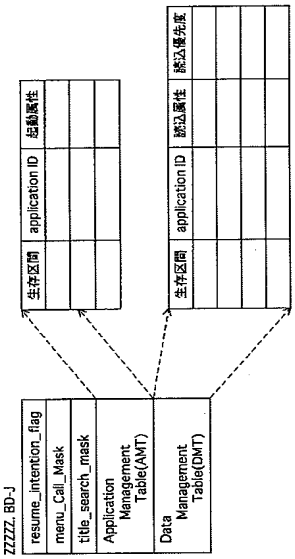
【図３７】



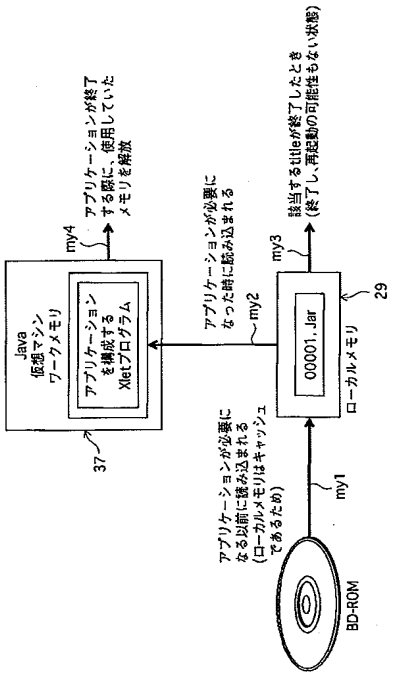
【図３８】



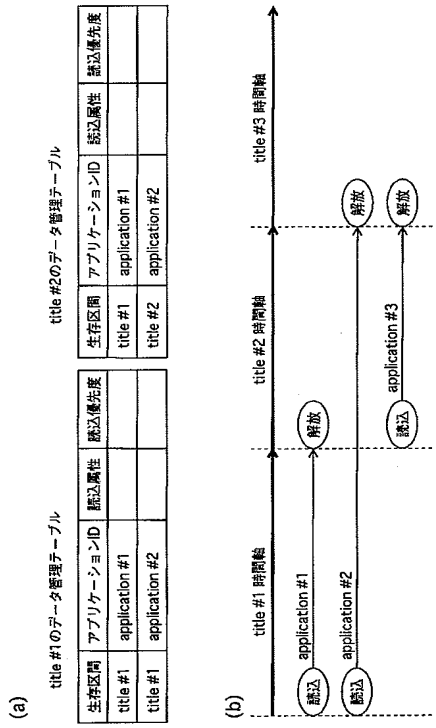
【図３９】



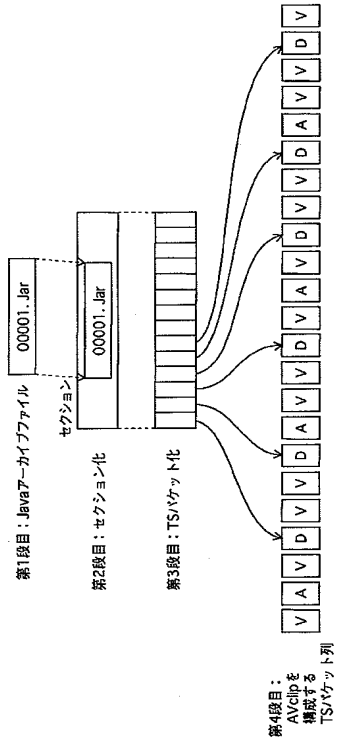
【図４０】



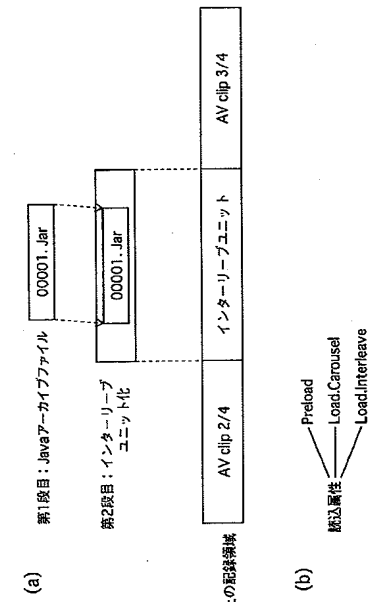
【図４１】



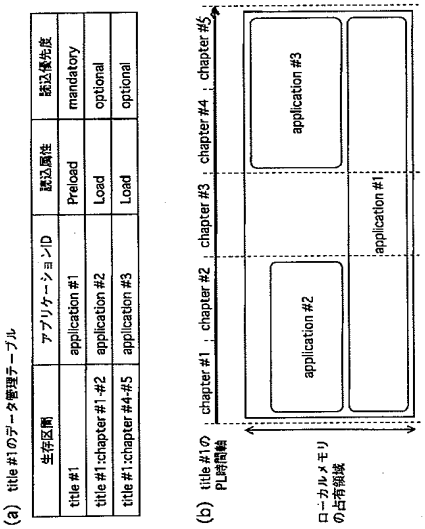
【図４２】



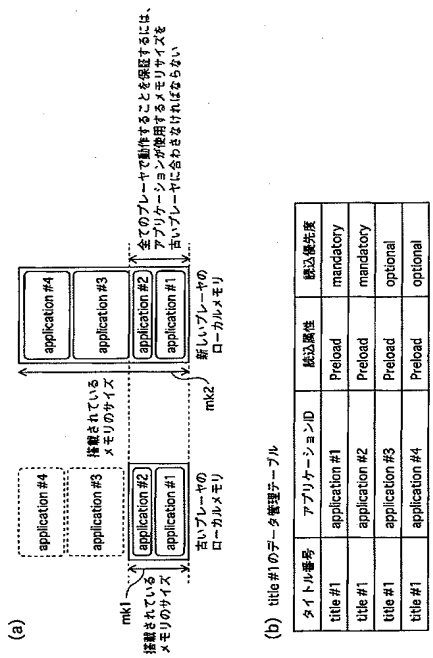
【図４３】



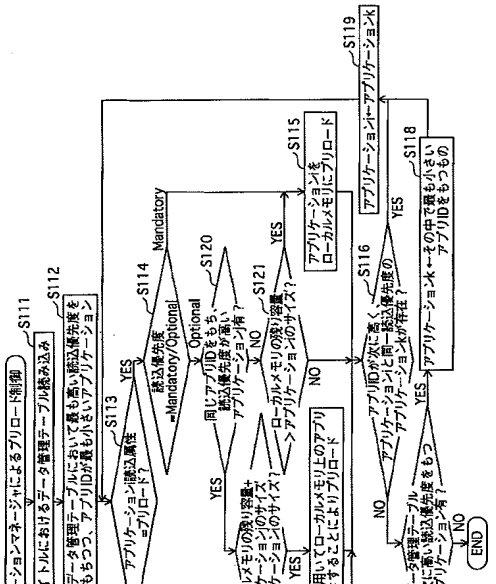
【図４４】



【図４５】



【図４６】

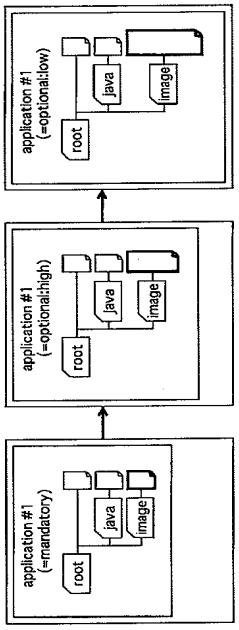


【図４７】

(a) title #1のデータ管理テーブル

生存区間	アプリケーションID	読込属性	読込優先度
title #1	application #1	Preload	mandatory
title #1	application #1	Preload	optional:high
title #1	application #1	Preload	optional:low

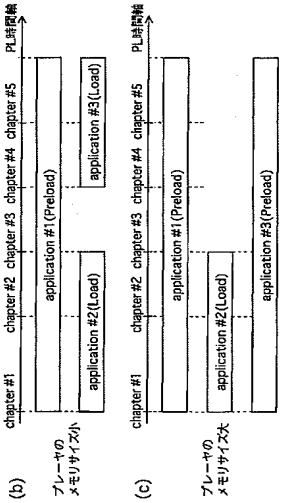
(b) ローカルメモリの占有領域



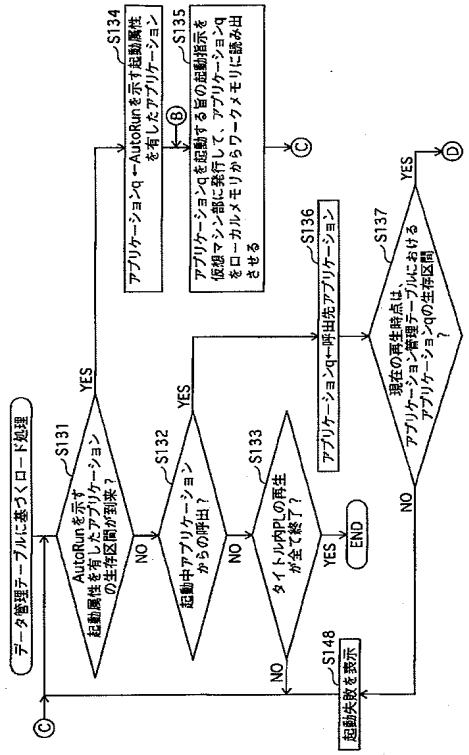
【図４８】

(a) title #1のデータ管理テーブル

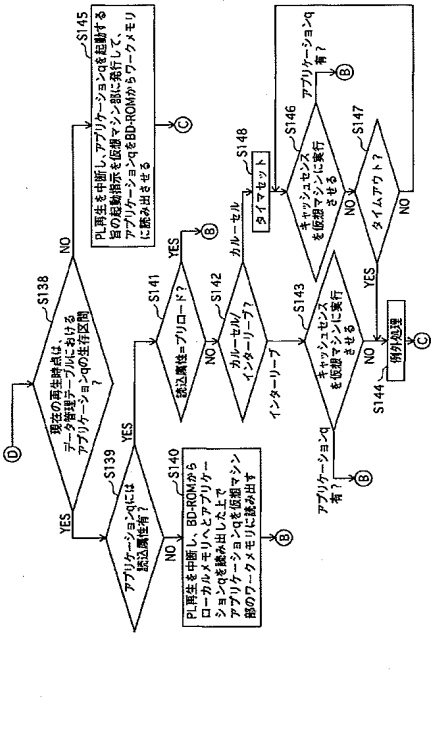
生存区間	アプリケーションID	読込属性	読込優先度
title #1	application #1	Preload	mandatory
title #1:chapter #1-#2	application #2	Load	mandatory
title #1:chapter #4-#5	application #3	Load	mandatory
title #1	application #3	Preload	optional



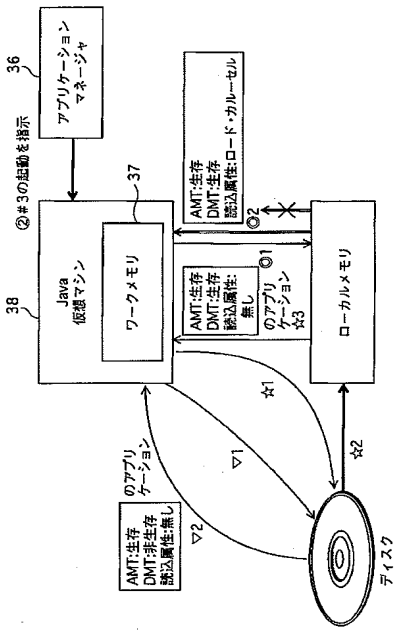
【図４９】



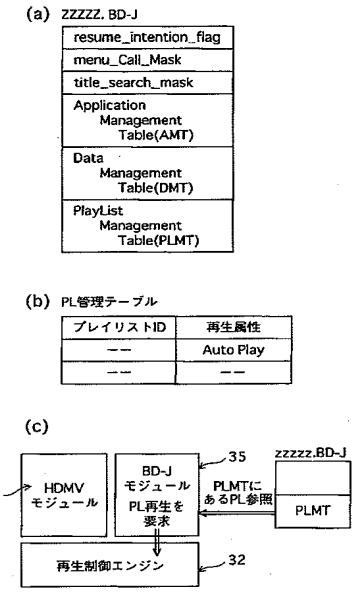
【図５０】



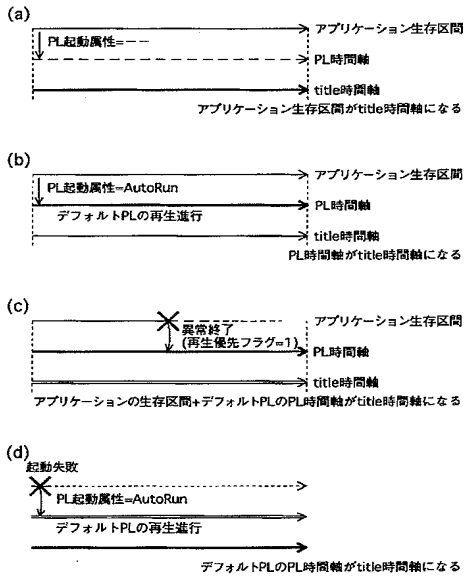
【図５１】



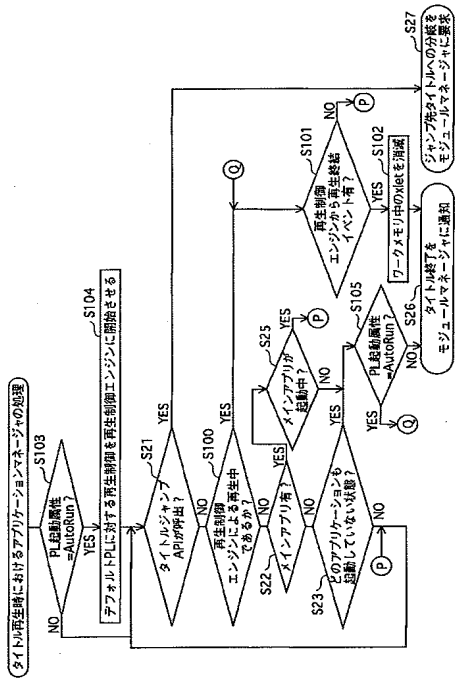
【図５２】



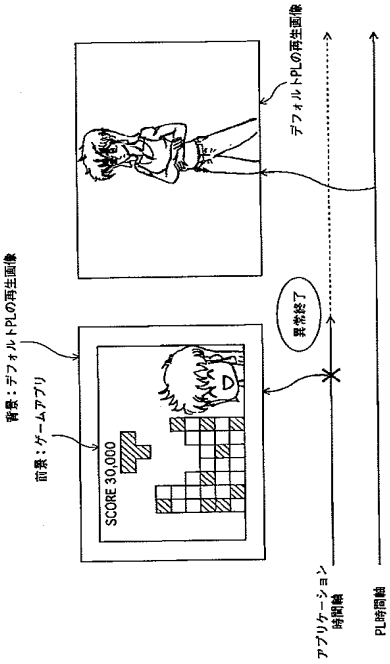
【図５３】



【図５４】



【図５５】



【図５６】

(a)

アプリケーション・データ管理テーブル

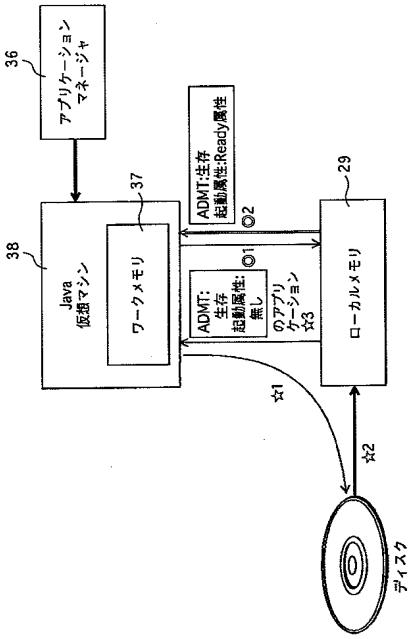
生存区間	アプリケーションID	起動属性	読込優先度
title #1	application #1	AutoRun	
title #1:chapter #1-#3	application #2	Ready	
title #1	application #3	--	
title #1:chapter #2-#4	application #4	Ready	

起動属性+読込属性

(b)

起動属性	AVストリーム再生前のアプリロード	自動起動/呼出起動	ローカルメモリへのロード	生存/非生存
AutoRun	○	自動	×	生存
AutoRun	×	自動	○	生存
READY	○	呼出	×	生存
READY	×	呼出	○	生存
無指定	×	呼出	×	生存

【図５７】

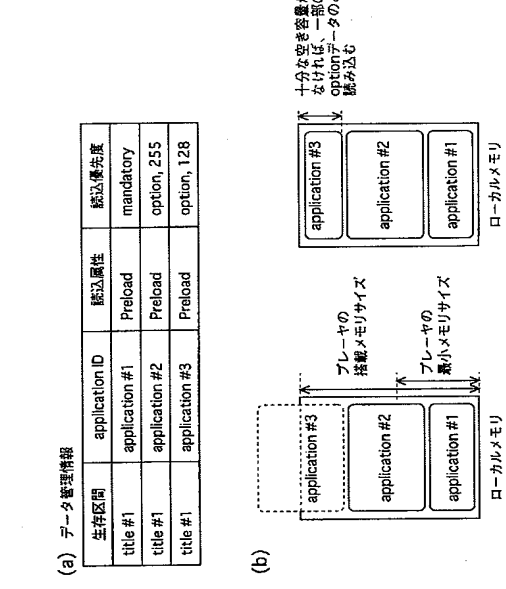


【図５８】

(a) データ管理情報

生存区間	application ID	読込属性	読込優先度
title #1	application #1	Preload	mandatory
title #1	application #2	Preload	option, 255
title #1	application #3	Preload	option, 128

(b)

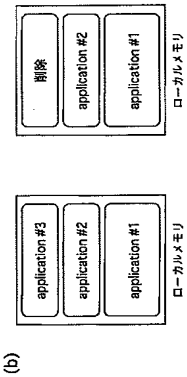


【図５９】

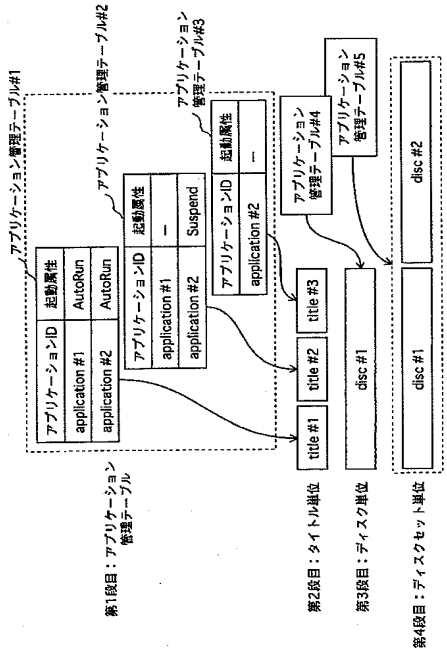
(a) データ管理テーブル

生存区間	アプリケーションID	ロード属性	読込優先度	グループ属性
title #1	application #1		mandatory	—
title #1	application #2		optional	group #1
title #1	application #3		optional	group #1

(b)



【図６０】



フロントページの続き

審査官　前田　祐希

(56)参考文献　国際公開第０１／０３５４１６（WO，Ａ１）
特開平０６－００４１６６（ＪＰ，Ａ）
特開平０６－２３０９４６（ＪＰ，Ａ）
特開平１１－２１９３１３（ＪＰ，Ａ）
特開２００２－３６９１５４（ＪＰ，Ａ）
特開２００３－０３２６３７（ＪＰ，Ａ）
特開２００２－２６９９２９（ＪＰ，Ａ）
特開２００１－０２２６２５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G11B　20/10
G11B　27/00
H04N　5/91-5/95